Bilkent University
Department of Computer Engineering

Senior Design Project
T2534
DriveMe

Analysis and Requirement Report

22102334, Eda Alparslan, eda.alparslan@ug.bilkent.edu.tr
22102720, Duru Solakoğlu, duru.solakoglu@ug.bilkent.edu.tr
21901440, Berfin Çetinkaya, berfin.cetinkaya@ug.bilkent.edu.tr
22001734, Ufuk Baran Guler, baran.guler@ug.bilkent.edu.tr
21601625, Ege Kaan Eren, kaan.eren@ug.bilkent.edu.tr

Prof. İbrahim Körpeoğlu
Mert Bıçakçı & İlker Burak Kurt

19.12.2025

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfilment of the requirements of the Senior Design Project course CS491/2.

Contents

List of Figures 5
1 Introduction 7
2 Current System 7
3 Proposed System 8
3.1 Overview 8
3.2 Functional Requirements 9
3.3 Non-functional Requirements 10
3.4 Pseudo Requirements 12
3.5 System Models 13
3.5.1 Scenarios 13
3.5.2 Use-Case Model 18
3.5.3 Object and Class Model 19
3.5.4 Dynamic Models 22
3.5.4.1 Activity Diagrams 22
3.5.4.2 State Diagrams 26
3.5.4.3 Sequence Diagrams 28
3.5.5 User Interface 29
4 Other Analysis Elements 41
4.1 Consideration of Various Factors in Engineering Design 41
4.1.1 Constraints 41
3.1.1.1 Developmental Constraints 42
3.1.1.2 Legal and Regulatory Constraints 42
3.1.1.3 Technological Constraints 43
3.1.1.4 Sustainability Constraints 43
4.1.2 Standards 43
4.2 Risks and Alternatives 44
4.3 Project Plan 45
4.4 Ensuring Proper Teamwork 52
4.5 Ethics and Professional Responsibilities 52
4.6 Planning for New Knowledge and Learning Strategies 53
5 Glossary 53
6 References 55

List of Figures

Fig. 1. Use Case Model..... 16
Fig. 2. Class Diagram..... 17
Fig. 3. Object Diagram 1..... 18
Fig. 4. Object Diagram 2..... 19
Fig. 5. Object Diagram 3..... 19
Fig. 6. Activity Diagram 1..... 20
Fig. 7. Activity Diagram 2..... 21
Fig. 8. Activity Diagram 3..... 22
Fig. 9. Activity Diagram 4..... 23
Fig. 10. State Diagram 1..... 24
Fig. 11. State Diagram 2..... 25
Fig. 12. Sequence Diagram 1..... 26
Fig. 13. Sequence Diagram 2..... 27
Fig. 14. Login screen..... 27
Fig. 15. Choosing Account type screen before preceding to registration page..... 28
Fig. 16. Registration screen for passenger..... 29
Fig. 17. Profile page for passenger..... 30
Fig. 18. Registration screen for driver..... 31
Fig. 19. Profile page for driver..... 32
Fig. 20. Homepage for passenger..... 32
Fig. 21. Homepage for driver..... 34
Fig. 22. Waiting offers screen for passenger..... 34
Fig. 23. Offers screen for passenger..... 35
Fig. 24. Waiting for driver screen for passenger..... 35
Fig. 25. Heading to Pickup screen for driver..... 35
Fig. 26. Verification screen for passenger..... 36
Fig. 27. Verification screen for driver..... 36
Fig. 28. Screen during trip for passenger..... 37
Fig. 29. Screen during trip for driver..... 37
Fig. 30. Screen after trip is completed for passenger..... 38
Fig. 31. Screen after trip is completed for driver/..... 39

List of Tables

TABLE I USE CASES..... 12
TABLE II FACTORS TABLE.....40
TABLE III RISK AND ALTERNATIVES..... 43
TABLE IV FACTORS..... 44
TABLE V LIST OF WORK PACKAGES.....46
TABLE IV DETAILED WPs.....47

Analysis and Requirement Report

DriveMe

1 Introduction

DriveMe is a mobile centric platform that connects licensed drivers with individuals who own a vehicle but prefer not to drive themselves. Instead of offering ride sharing or taxi like services, DriveMe enables passengers to request a professional driver to operate the passenger's own car for a trip. The platform is designed to meet the unique constraints of Turkish transportation law, which prohibits informal ride sharing or peer to peer commercial passenger transport. By matching a vetted driver to a passenger's vehicle, the system provides convenience and peace of mind while maintaining legal compliance.

This report presents the analysis and requirements for the DriveMe system. It outlines the functional and non functional specifications derived from stakeholder needs and legal constraints, describes the overall system architecture, and identifies the assumptions and constraints that shape design decisions. The aim is to ensure that the project's scope, behaviour and quality attributes are clearly defined to guide development and testing.

2 Current System

Today, users who need a driver for their own car can use a few specialized services, but these are not widely available everywhere. For example, Dryver provides an app where users can book a professional driver to drive their own vehicle on demand. Users can make reservations, track the driver, and pay through the app while staying in their car throughout the trip [1].

Another example is Get Driven, an app that allows users to request a trained driver for their own car. It offers real-time driver assignment, driver profiles, and in-app payment [2].

Traditional chauffeur and private driver services also exist in some regions. In Türkiye, there are companies that offer chauffeur-driven car services, where customers hire a driver and vehicle for travel, transfers, or tours. These services

operate through booking platforms, but they do not focus on on-demand drivers for users' own cars in everyday use [3].

Most global mobility apps like Uber or DiDi focus on transporting passengers in the driver's car, not driving the user's own vehicle. DiDi does have a designated driving service in some regions that drives the customer's car, but it is not available everywhere [4].

In summary, there are some existing services that provide drivers for users' vehicles, but these options are limited in availability and scope. DriveMe aims to expand this model with a user-friendly, widely accessible on-demand driver service for personal cars.

3 Proposed System

3.1 Overview

DriveMe consists of a front end mobile application (Android) and a back end service that handles user management, matching, pricing and trip tracking. Users can sign up as either passengers or drivers. Passengers register their personal details and vehicle documentation; drivers register their licence and identity documents. An administrator verifies both vehicle and driver records. Once verified, passengers can post ride requests by providing their vehicle, current location and destination. The system's matching engine recommends a price range based on distance, duration and current demand. Licensed drivers view nearby pending requests, submit offers within the suggested range and receive notifications when a passenger accepts their offer.

Prior to departure, passenger and driver authenticate each other via a QR code generated by the system. During the trip, the application activates GPS tracking to monitor route adherence. Upon arriving at the destination, the passenger ends the ride, and both parties confirm that the trip was completed and the driver received cash payment. Passengers can then rate and review drivers. Administrators oversee user registration, verify documentation and handle disputes. The system relies on third party services such as Google Maps API for geolocation and routing, and must comply with Turkey's Personal Data Protection Law (KVKK) and the General Data Protection Regulation (GDPR).

3.2 Functional Requirements

Functional requirements specify the behaviours and operations the system must perform to satisfy business and user needs. They define what the system should do and how it should interact with users or other systems. For DriveMe, the core functional requirements are derived from the use-case scenarios. They are numbered for traceability.

FR-1: The system shall allow new users to sign up by entering personal information, verifying that the email or phone number is unique and that the user is of legal age.

FR-2: The system shall authenticate users via username/password and, where necessary, two-factor authentication, allowing them to log in and log out securely.

FR-3: The system shall allow users to edit their profile information and reset their password after logging in.

FR-4: The system shall allow passengers to register their vehicles by uploading relevant documents for administrator approval; vehicles shall carry statuses such as “WAITING APPROVAL”, “VERIFIED” or “REJECTED”.

FR-5: The system shall allow drivers to register as licensed drivers by uploading their licence and identity documents for verification.

FR-6: The system shall allow verified passengers to create a ride request by specifying the vehicle, current location and destination; the system shall display an estimated price range.

FR-7: The system shall allow drivers to search for and view pending ride requests within a suitable radius, displaying details such as destination, vehicle type and price range.

FR-8: The system shall allow drivers to submit a price offer within the recommended range for a selected ride request.

FR-9: The system shall allow passengers to view offers for their requests and accept one offer, thereby matching the passenger and driver and flagging the request as “MATCHED”.

FR-10: The system shall generate a unique QR code for each matched trip; scanning the code at the departure location shall identify both parties and flag the request as “TRIP STARTED”.

FR-11: The system shall provide real-time GPS tracking during the trip and log the route, supporting safety monitoring and dispute resolution.

FR-12: The system shall allow passengers to end the trip; it shall verify that the passenger is at the destination, prompt the driver to confirm receipt of cash payment and flag the ride as “COMPLETED”.

FR-13: The system shall allow passengers to rate drivers after trip completion and display cumulative ratings on driver profiles.

FR-14: The system shall notify drivers when their offer is accepted, when a trip has started and when it has ended.

FR-15: The system shall allow administrators or designated managers to review and approve or reject vehicle and driver documentation.

FR-16: The system shall provide a pricing engine that recommends an appropriate price range based on factors such as distance, time and demand.

FR-17: The system shall log the value of cash transactions and maintain audit records of completed trips for accountability.

FR-18: The system shall support secure session management, preventing unauthorized access to user accounts.

3.3 Non-functional Requirements

Non-functional requirements define how the system should behave; they describe quality attributes such as performance, reliability and security rather than specific features. The following non-functional requirements apply to DriveMe.

Performance: The mobile application should respond to user actions (login, creating a request, searching for offers) within seconds under normal conditions.

Availability: The system should achieve 99 % uptime and gracefully handle partial outages, ensuring that pending requests are cached locally so that drivers can still view them during brief connectivity issues.

Scalability: The system should be horizontally scalable to accommodate increases in the number of users, vehicles and drivers. It should allow database and server resources to be expanded without redesigning the application.

Security and Privacy: All personal data shall be stored securely using encryption. The system must comply with the Turkish Personal Data Protection Law (KVKK) and GDPR, collecting only necessary data, storing location data for no longer than operationally required and protecting it from unauthorized access.

Reliability: The application should recover gracefully from network interruptions. Local caching of pending requests and robust retry mechanisms shall prevent data loss and ensure continuity.

Maintainability: The code base shall be modular and well documented to simplify updates and bug fixing. The design should support clear separation of concerns among user management, matching, pricing and tracking modules.

Usability: The user interface shall be intuitive and accessible, with clear navigation and feedback. The app should be usable by elderly and disabled users, for example through appropriate font sizes and contrast.

Compatibility: The mobile application shall support Android version 8.0 or higher on devices with at least 4 GB of RAM.

Dependence on Third-Party Services: The system shall use reliable third-party services such as Google Maps API for geolocation and routing. It should handle API limitations, outages or pricing changes by providing fallback options or notifications.

Compliance with Legal Constraints: The system shall restrict functionality to driver services where a licensed driver operates the user's car. It shall not support shared rides, pooled trips or taxi-like operations, ensuring compliance with Turkish transportation regulations.

Data Integrity: All changes to trip states (requested, matched, started, completed) shall be transactional so that incomplete operations do not leave the system in an inconsistent state.

3.4 Pseudo Requirements

Pseudo requirements capture constraints, standards and design decisions that influence implementation but do not directly describe system behaviour. They serve as guidelines for development and project management.

• 1. The mobile front end shall be implemented using native cross platform framework such as Flutter, depending on team expertise.

• 2. The back end shall be developed with a modern web framework (Spring Boot) following RESTful API principles.

• 3. The project shall use Git for version control and adhere to a structured branching model to support collaborative development.

• 4. Continuous integration and delivery (CI/CD) pipelines shall be established to automate building, testing and deployment.

• 5. All third party libraries and frameworks shall be properly licensed and compatible with the project's licensing strategy.

• 6. Development and testing shall follow an agile methodology with regular sprint reviews and retrospectives to incorporate feedback and adjust scope.

• 7. Software documentation shall comply with IEEE 1471 for architectural descriptions and UML notation for system modelling.

3.5 System Models

3.5.1 Scenarios

TABLE I

USE CASES

Use Case #1	Requesting a ride.
Actor	Passenger
Flow of Events	1. Passenger logs in to the application and confirms their location. 2. They choose the location they want to travel to, and the type of vehicle that they own. 3. Application displays the estimated fee interval.
Entry Condition	● Passenger is logged in. ● The passenger has a recorded and verified vehicle. ● The destination point is chosen.
Exit Condition	● Passenger completes the request.

Use Case #2	Offering for a specific ride request.
Actor	Driver
Flow of Events	1. The driver logs in and searches for the nearby driver requests. 2. The driver offers a price within the price range that the application calculates.
Entry Condition	● The driver is logged in. ● The driver holds a valid driver's license for the specific request.
Exit Condition	● The driver submits the offer.

Use Case #3	Matching requests and offers.
Actor(s)	Passenger & Driver
Flow of Events	1. The passenger views the offers for chosen requests. 2. The passenger accepts one of the offers that they see fit.

	3. The driver gets notified that their offer is accepted.
Entry Condition	● Passenger is logged in. ● The specific request has received at least one offer. ● The specific request is not withdrawn.
Exit Condition	● The match is made and the request is flagged as “MATCHED”

Use Case #4	Initiating a trip.
Actor(s)	Passenger & Driver
Flow of Events	1. The passenger and the driver arrive at the chosen departure point. 2. They recognize each other by scanning the 4-digit code that the app provides. 3. The request is flagged as “TRIP STARTED” when the 4-digit code gets scanned.
Entry Condition	● Passenger and driver arrive at the departure point. ● The 4-digit code is generated by application and recognized.
Exit Condition	● The 4-digit code is scanned.

Use Case #5	Completing a trip.
Actor	Passenger & Driver
Flow of Events	1. Passenger logs in to the app. 2. The passenger pushes the relevant button to “End Ride”. 3. The application checks their location and compares it to the destination point. 4. The driver is notified and also asked to approve. 5. The driver should also approve that they have received the payment in cash. 6. The ride is flagged as “COMPLETED”.
Entry Condition	● The ride is flagged as “TRIP STARTED”.
Exit Condition	● The driver approves that they received the

	payment.

Use Case #6	Rating a trip.
Actor	Passenger
Flow of Events	1. Passenger logs in to the application and chooses one of their past trips. 2. The passenger is asked to rate the driver in some aspects. 3. The passenger submits their review. 4. The rating is displayed on the driver's profile.
Entry Condition	● Passenger is logged in. ● The passenger has completed at least one trip before.
Exit Condition	● Passenger submits the rating.

Use Case #7	Register a vehicle.
Actor	Passenger
Flow of Events	1. Passenger logs in to the application. 2. They navigate to the "Add Vehicle" page. 3. The passenger uploads relevant documentation. 4. The vehicle gets flagged as "WAITING APPROVAL". 5. An administrator approves the documents. 6. The vehicle is registered and unflagged.
Entry Condition	● Passenger is logged in. ● The passenger is able to provide relevant documentation.
Exit Condition	● Passenger submits the registration.

Use Case #8	Register as a licensed Driver.
Actor	Driver

Flow of Events	1. The driver logs in to the application. 2. They navigate to the “Get Verified” page. 3. The driver uploads relevant documentation. 4. Their profile gets flagged as “WAITING APPROVAL”. 5. An administrator approves the documents. 6. The driver is registered and unflagged.
Entry Condition	● The driver is logged in. ● The driver is able to provide relevant documentation.
Exit Condition	● The driver submits the registration.

Use Case #9	Login
Actor	User (Not logged-in)
Flow of Events	1. The user opens the application. 2. Enters valid credentials and completes 2FA if necessary. 3. The login button is clicked.
Entry Condition	● The user is not logged in. ● The application is installed in the device. ● The device is connected to the internet.
Exit Condition	● The user is logged in.

Use Case #10	Sign up
Actor	Non-User
Flow of Events	1. Non-user installs the application. 2. They enter their personal information. 3. Their e-mail or phone number is checked for duplicates. 4. They create a password. 5. They submit the sign up form.
Entry Condition	● The non-user has no other account. ● The non-user is of legal age.
Exit Condition	● The new user gets registered.

Use Case #11	Edit profile
Actor	User
Flow of Events	1. The user navigates to the Edit Profile page. 2. They enter the new credentials/information that they want updated. 3. They submit the form. 4. The database gets updated with new information.
Entry Condition	● The user is logged in.
Exit Condition	● The user leaves the Edit Profile page.

Use Case #12	Reset Password
Actor	User
Flow of Events	5. The user navigates to the Reset Password page. 6. They enter the current and the new password. 7. They submit the form. 8. The database gets updated with new information.
Entry Condition	● The user is logged in.
Exit Condition	● The user leaves the Reset Password page.

3.5.2 Use-Case Model

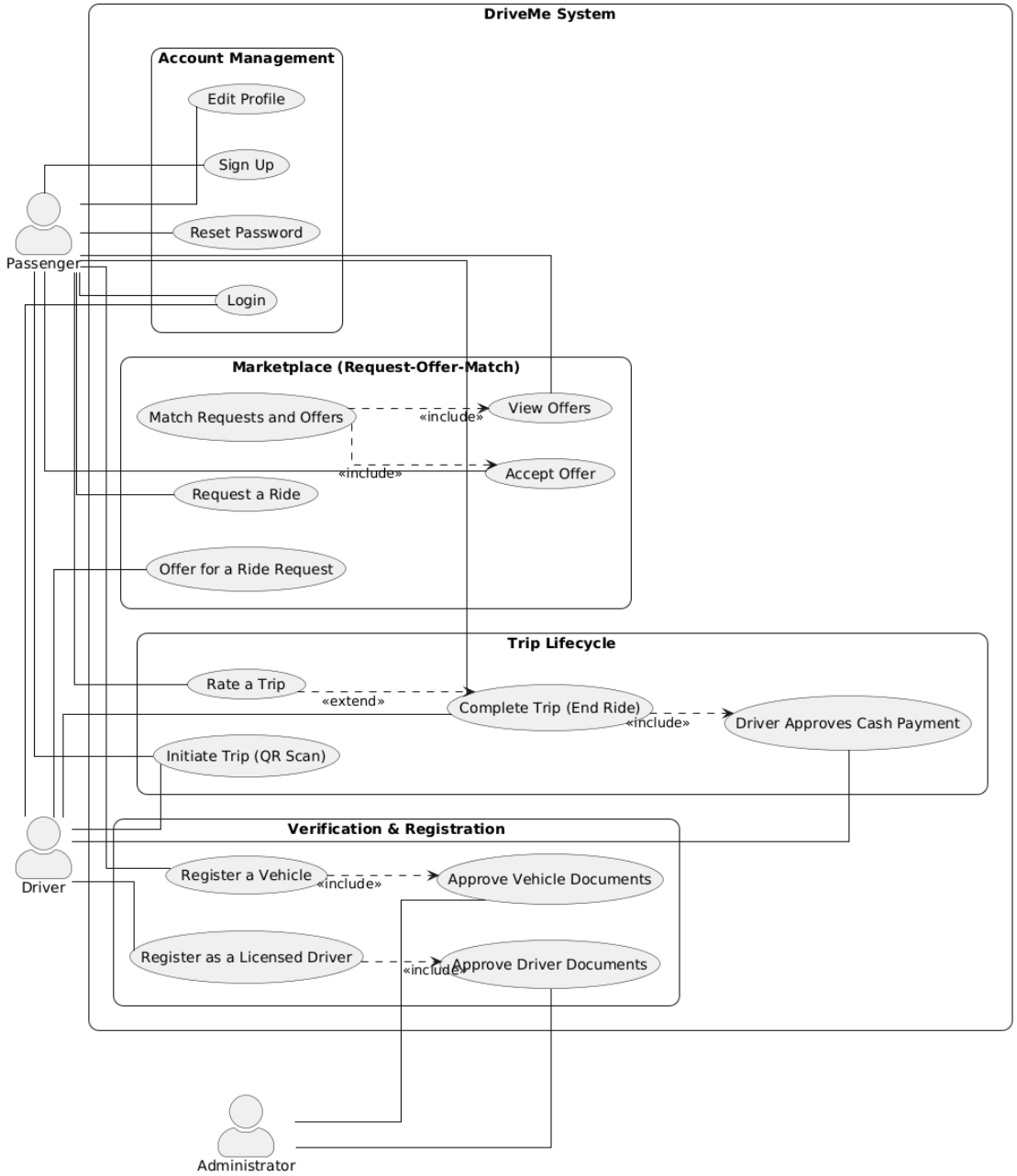


Fig. 1. Use Case Model.

3.5.3 Object and Class Model

3.5.3.1 Class Diagram

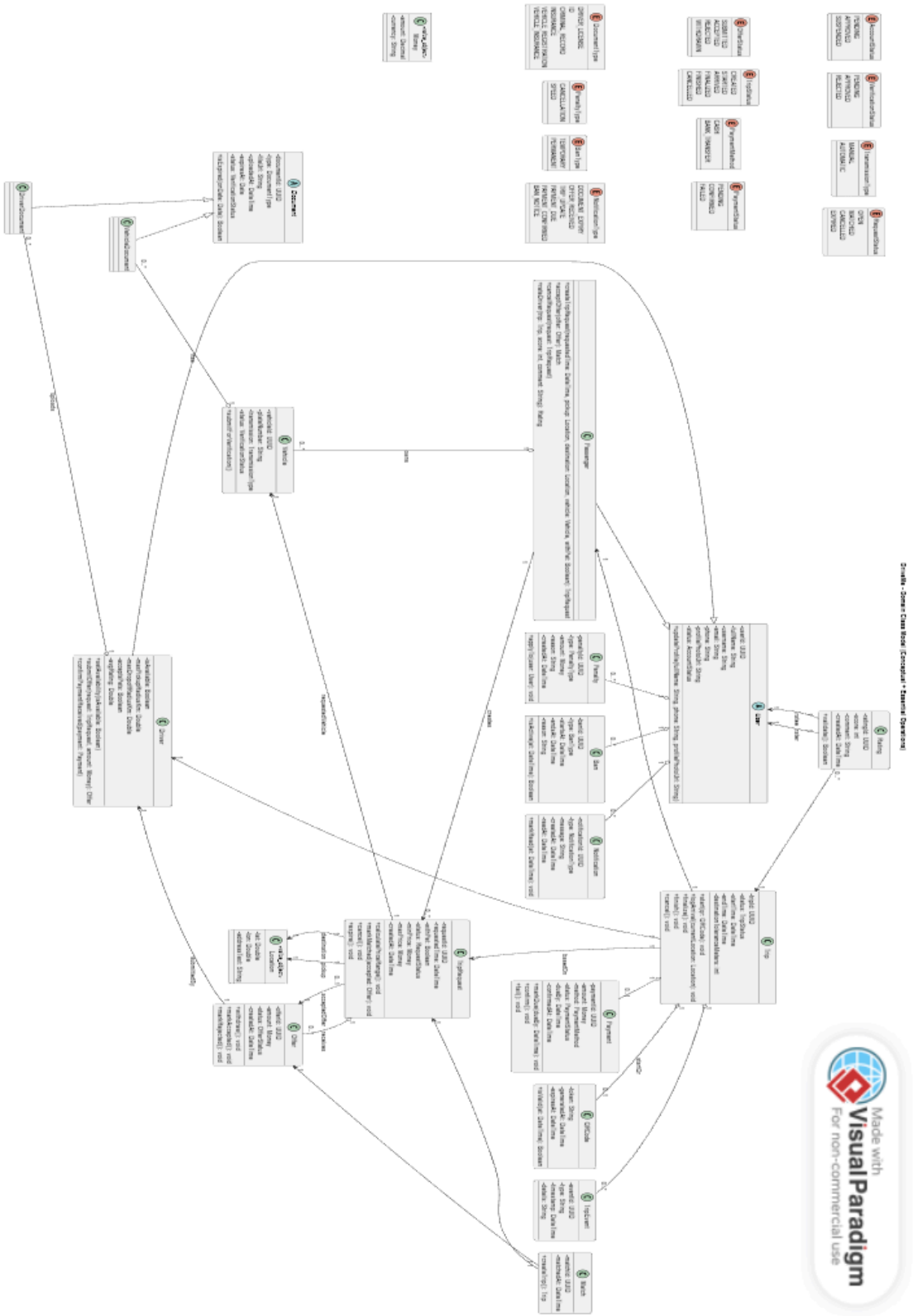


Fig. 2. Class Diagram.

3.5.3.2 Object Diagrams

This section includes object diagrams

3.5.3.2.1 Matching created, Trip started, Payment pending

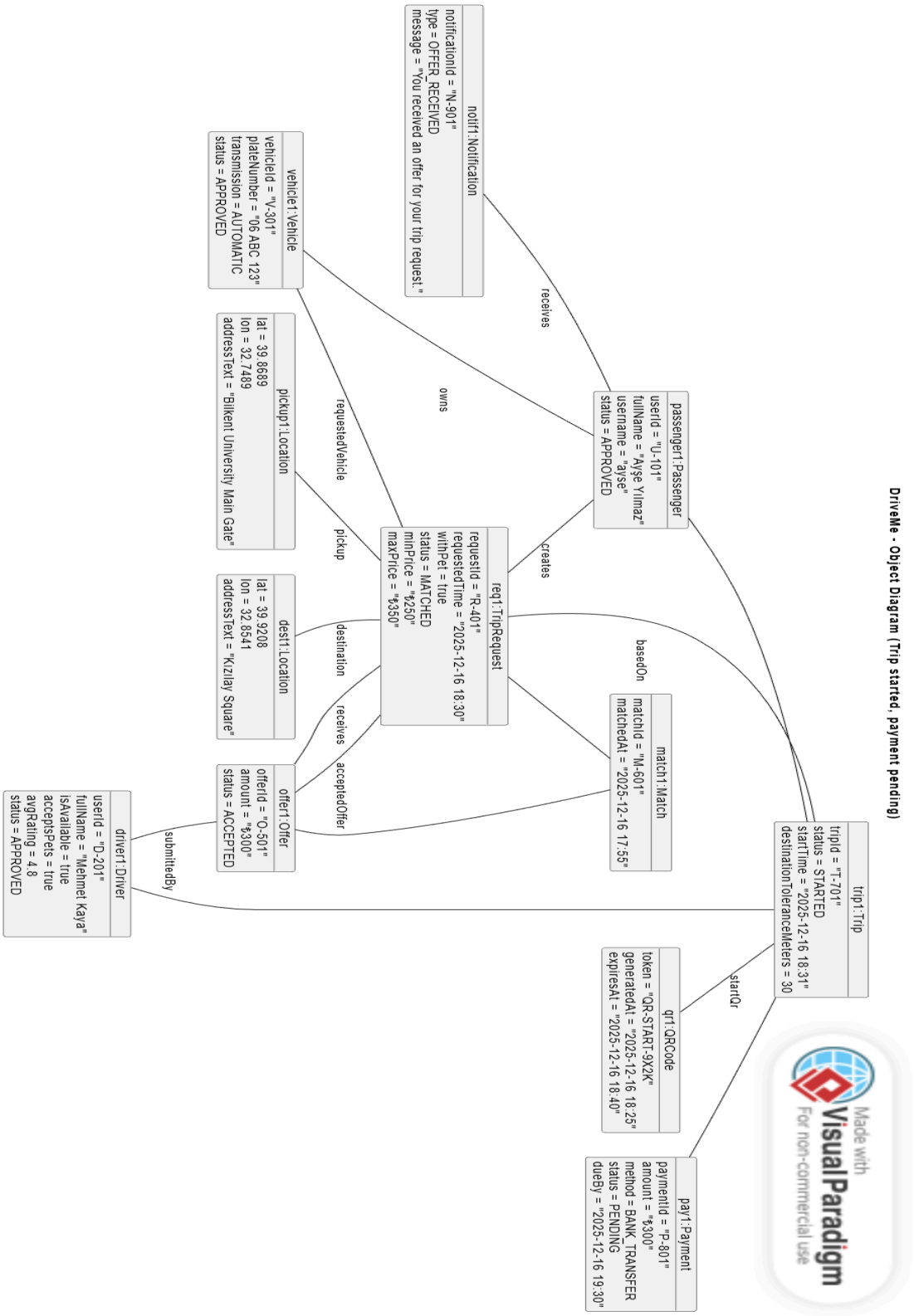


Fig. 3. Object Diagram 1.

3.5.3.2.2 Trip finalized, payment confirmed, rating given

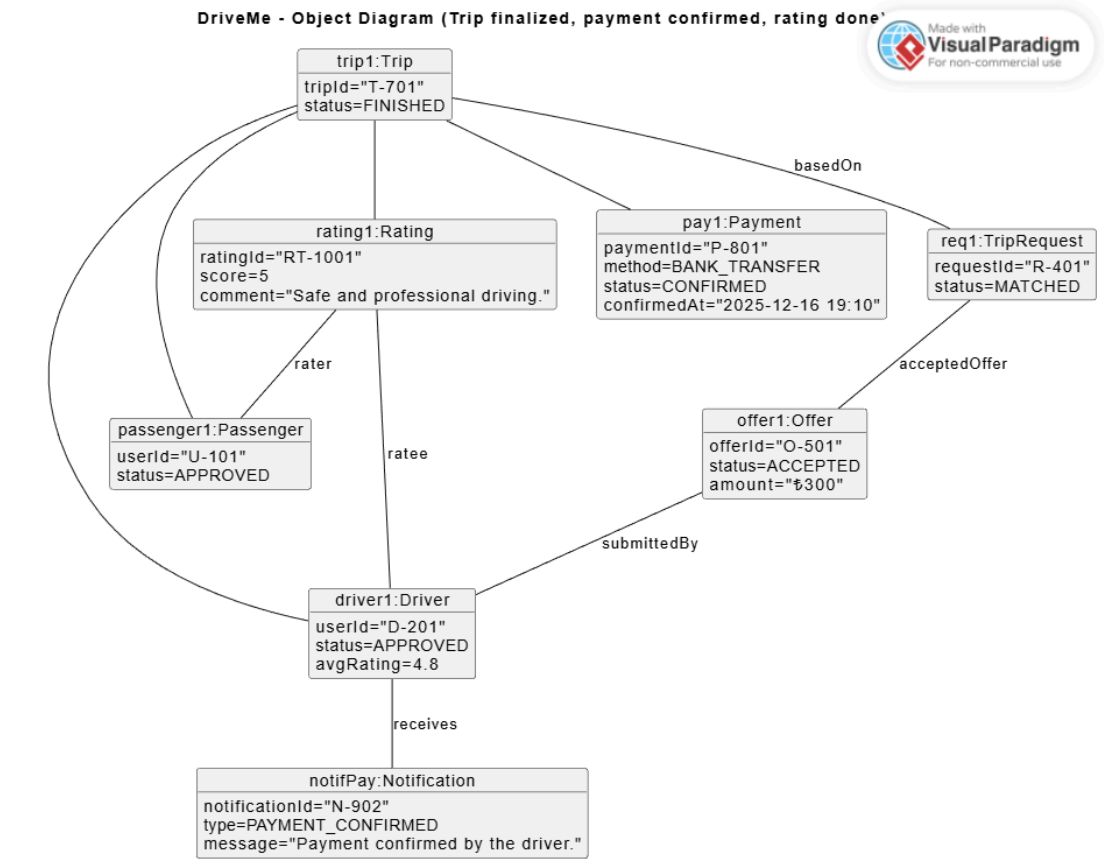


Fig. 4. Object Diagram 2.

3.5.3.2.3 Document expired → notification + account issue (compliance snapshot)

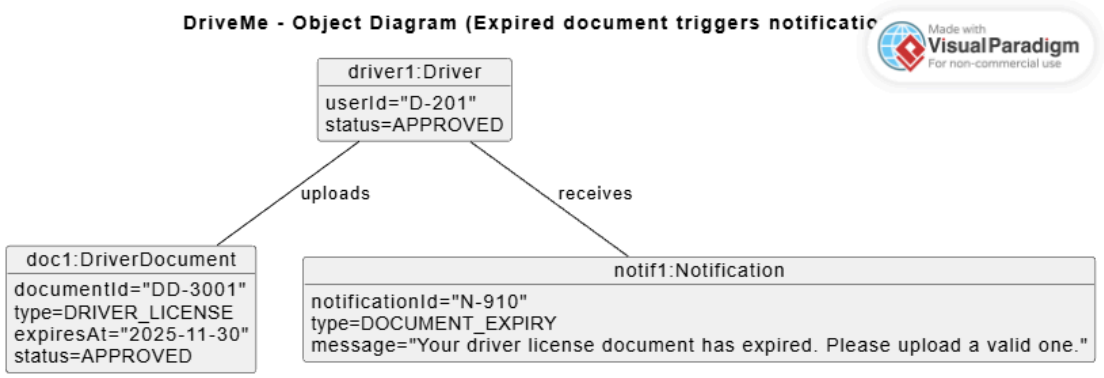


Fig. 5. Object Diagram 3.

3.5.4 Dynamic Models

3.5.4.1 Activity Diagrams

User Registration and Verification: User Registration and Verification Activity Diagram explains how a new passenger or driver joins the platform. It details the process of uploading documents where drivers are granted licenses for mandatory verification by the Security Manager. This flow ensures that all users meet the system's security and identity standards before interacting with the marketplace.

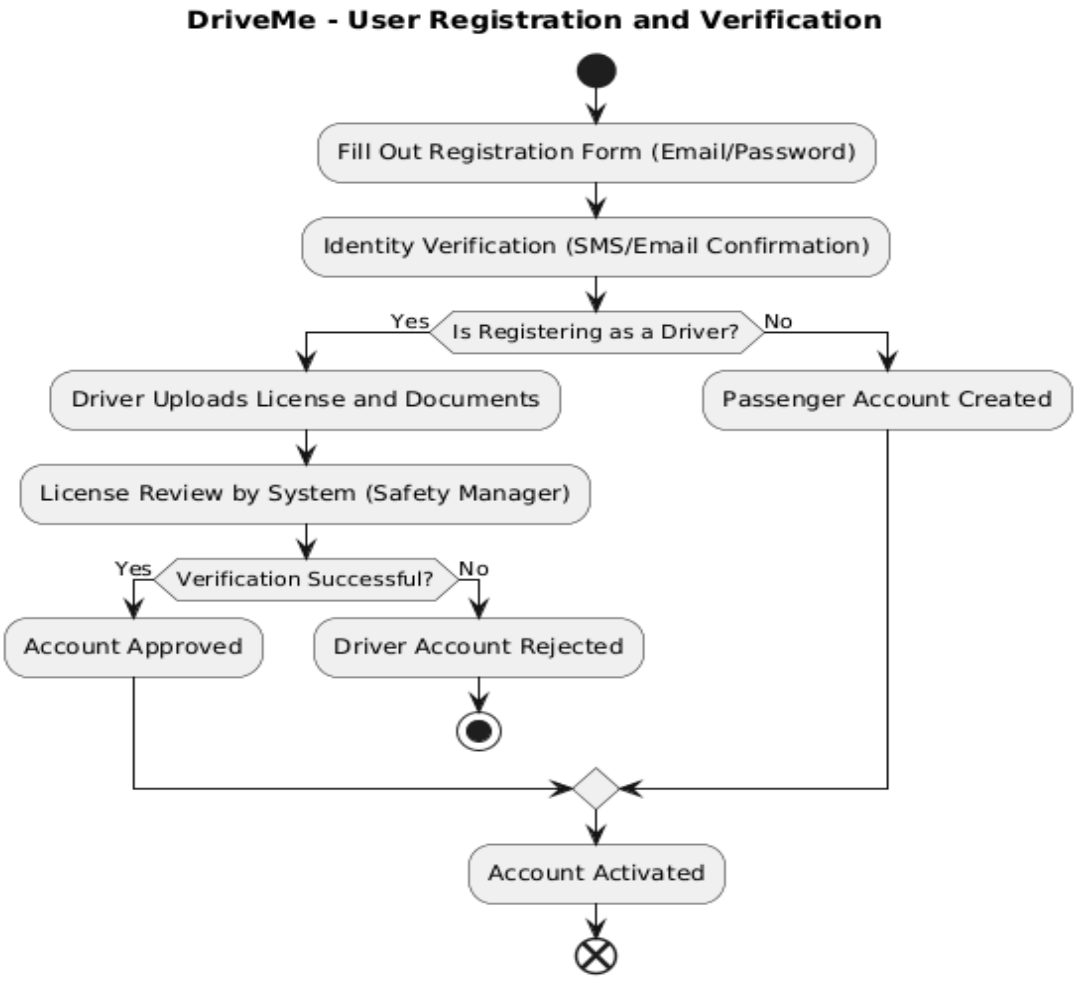


Fig. 6. Activity Diagram 1.

Trip Request Creation: This diagram shows how individual drivers initiate the matching process by viewing pending requests. It explains how the Pricing Engine provides intelligent price recommendations, which the driver uses to send a personalized offer to the passenger. The process ends when the passenger reviews the offer and accepts it to finalize the travel contract.

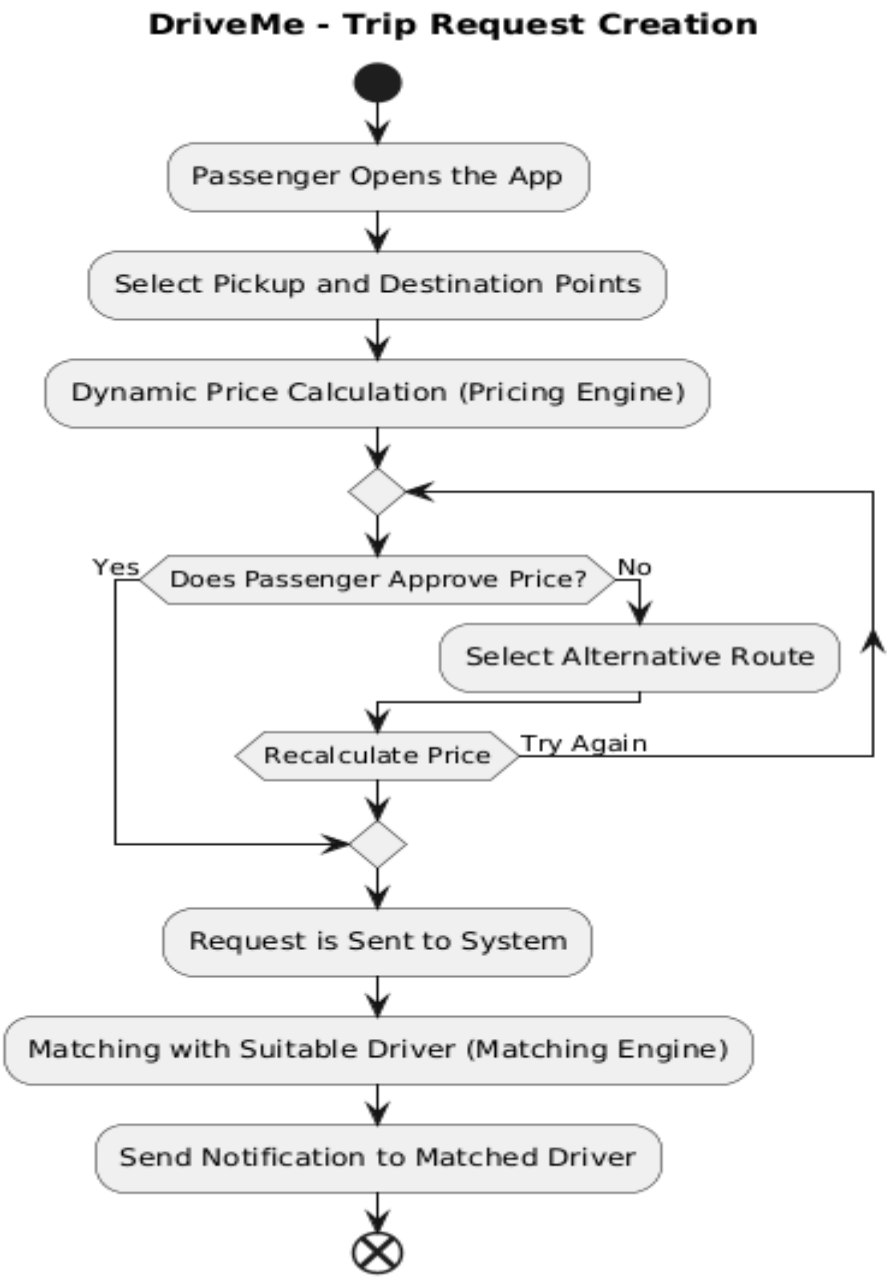


Fig. 7. Activity Diagram 2.

Driver Accepts and Proceeds: This diagram details the driver's journey to the owner defined pick-up point. It emphasizes the use of the Google Maps API for real time navigation and arrival notifications. This allows the driver to reach the passenger's location efficiently once the offer has been successfully accepted.

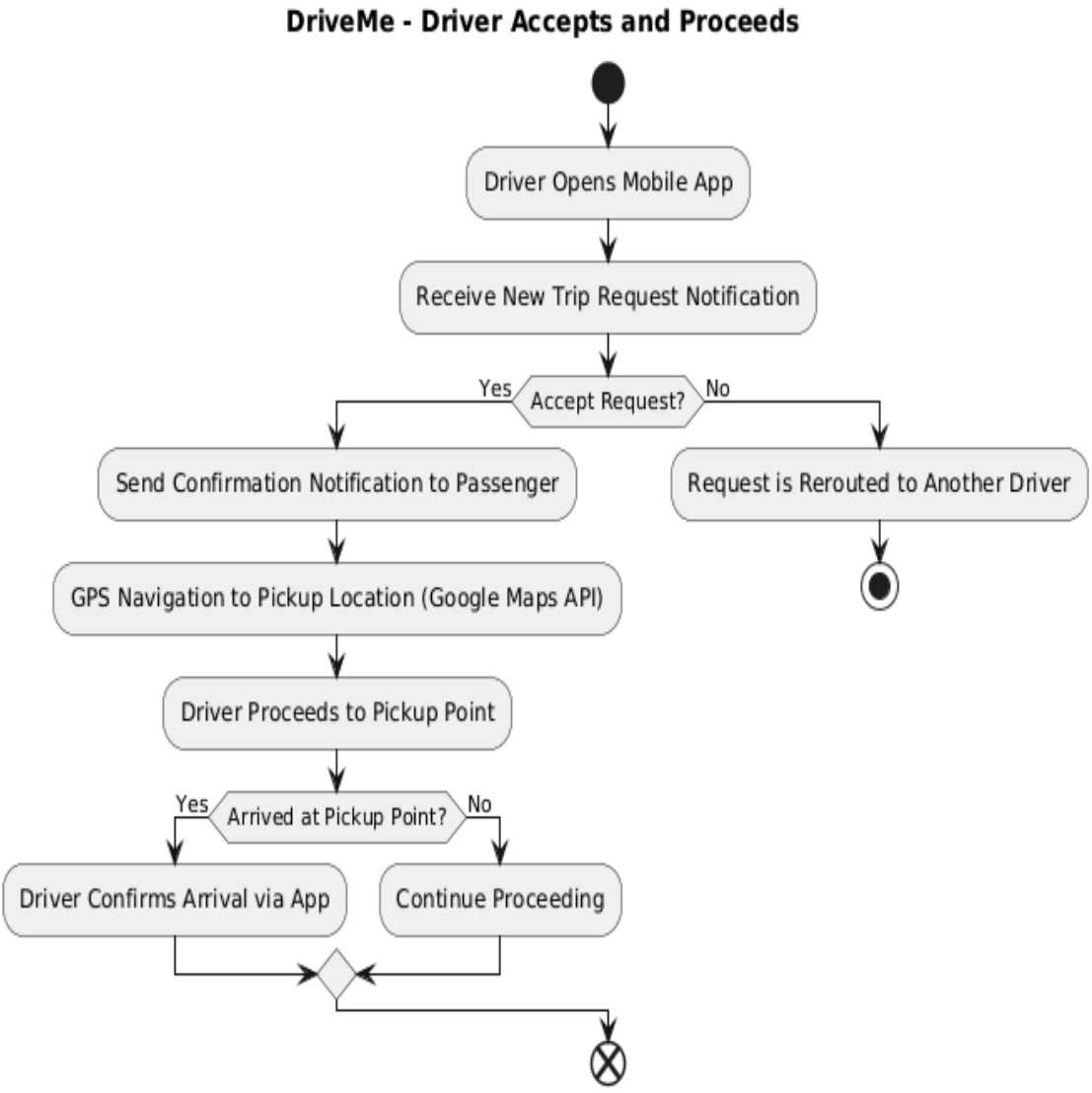


Fig. 8. Activity Diagram 3.

Trip Initiation and Tracking: This diagram explains the moment the driver takes control of the owner's vehicle. It covers the activation of GPS tracking by the Safety Manager to monitor the trip in real time. This stage focuses on passenger safety and adherence to the defined route throughout the journey.

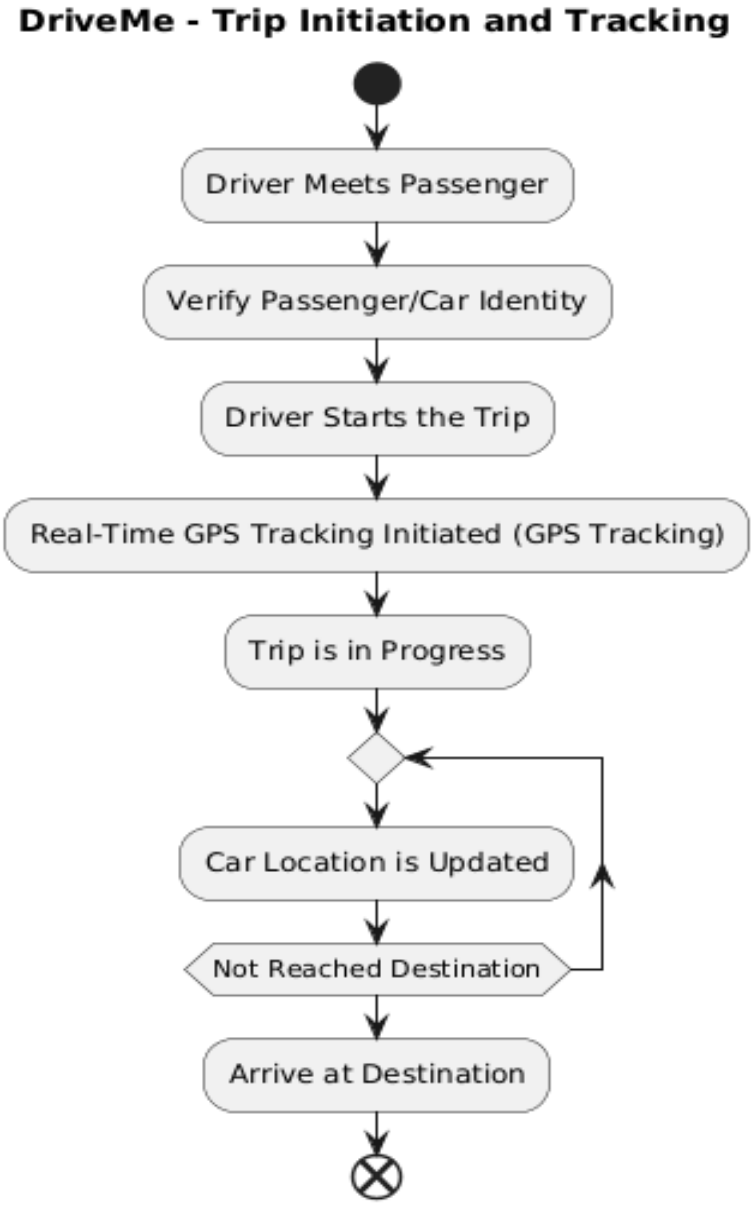


Fig. 9. Activity Diagram 4.

3.5.4.2 State Diagrams

Verification and Payment: This Diagram manages the hiring process to ensure drivers meet licensing requirements. No driver can participate in the tender process until their documents have been "Verified" by the Safety Manager. This ensures that security restrictions and ethical data processing rules are adhered to.

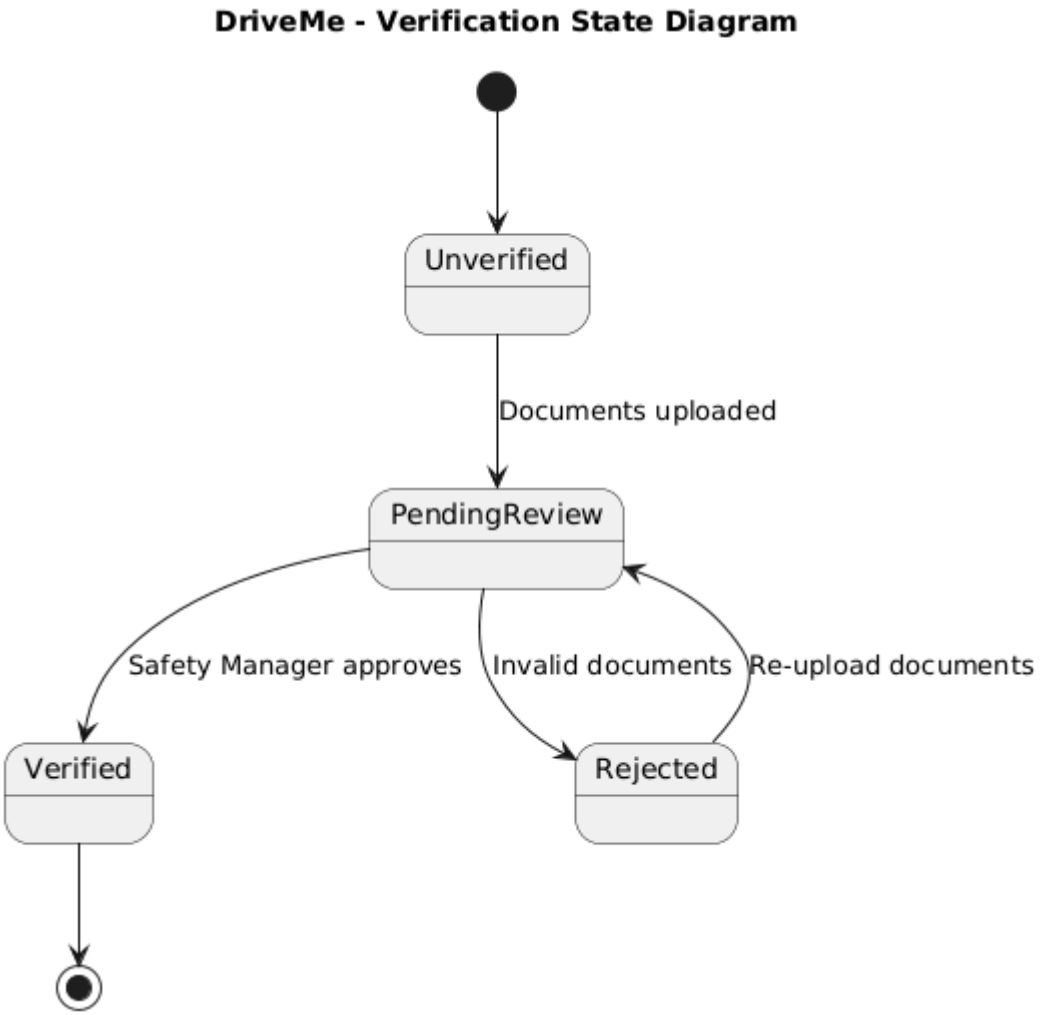


Fig. 10. State Diagram 1.

Trip LifeCycle: The Trip State Diagram tracks the entire journey of a service request from creation to completion. It begins when a passenger posts a request and transitions through driver bidding, active driving with GPS tracking, and final arrival.

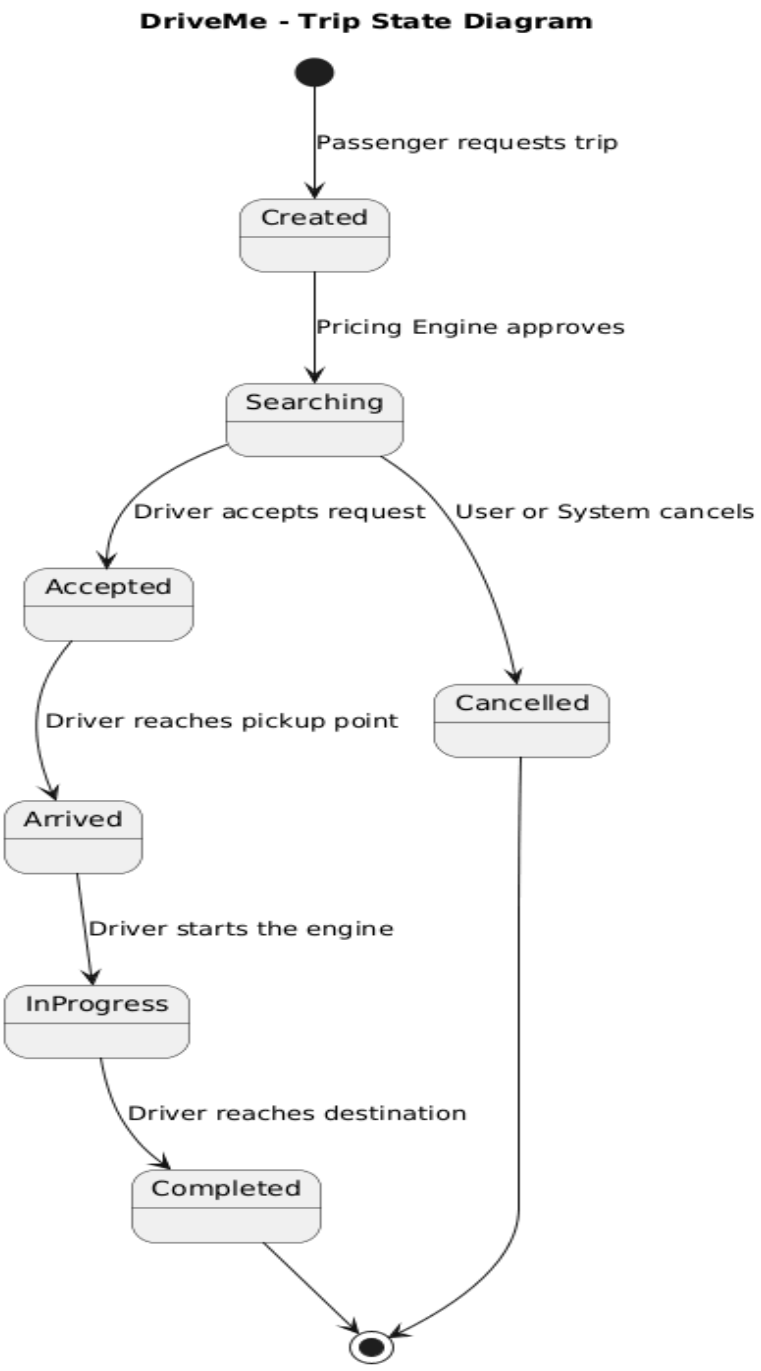


Fig. 11. State Diagram 2.

3.5.4.3 Sequence Diagrams

Matching and Bidding: This diagram shows the interaction between the driver, the matching engine, and the passenger, where the driver initiates the matching by sending a quote for a pending request. It explains how the pricing engine intelligently suggests a price range to the driver, presenting a custom offer that the driver must then confirm to finalize the passenger's trip contract.

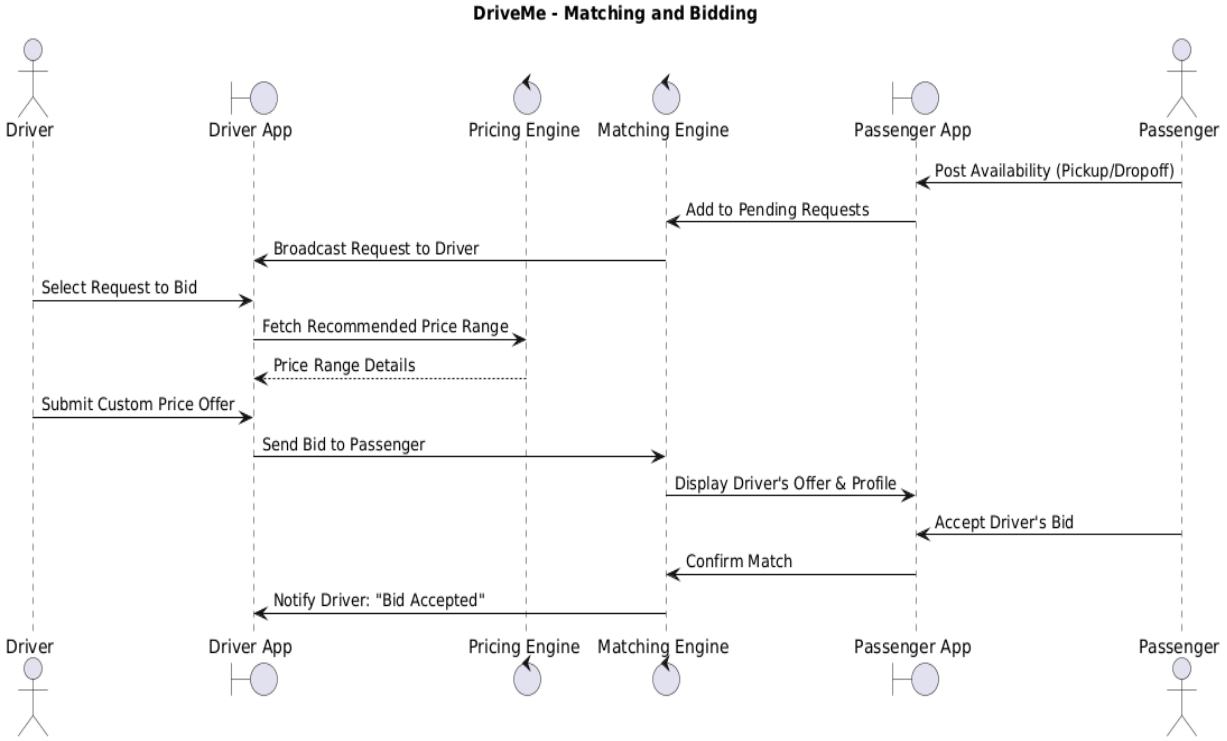


Fig. 12. Sequence Diagram 1.

Trip Finalization Sequence: This diagram shows the interaction between the Driver, Executive Manager and Payment Manager during the termination of a service.The process ends with the Payment Manager collecting the funds and the Traveler receiving a digital receipt for the completed service.

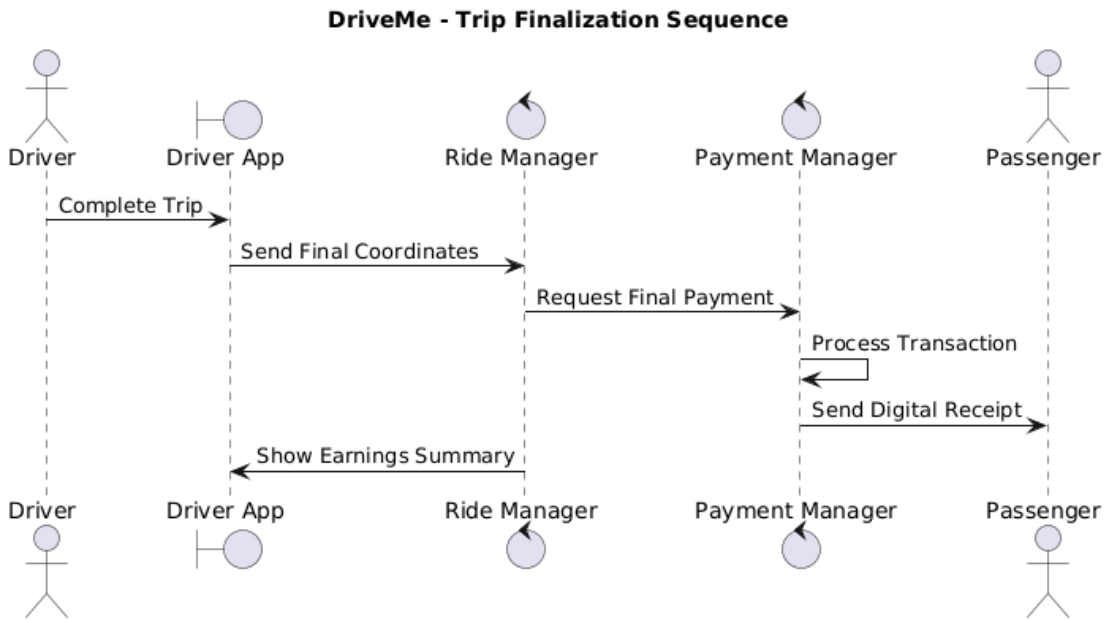


Fig. 13. Sequence Diagram 2.

3.5.5 User Interface

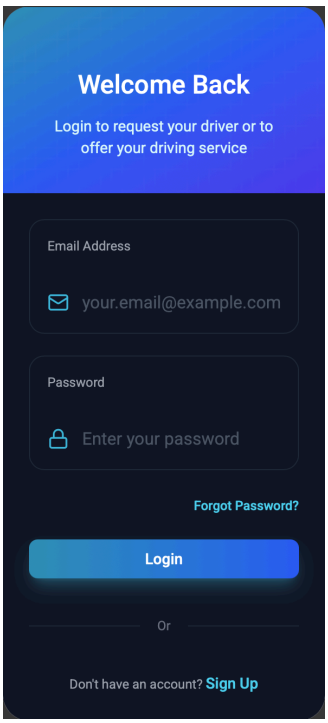


Fig. 14. Login screen.

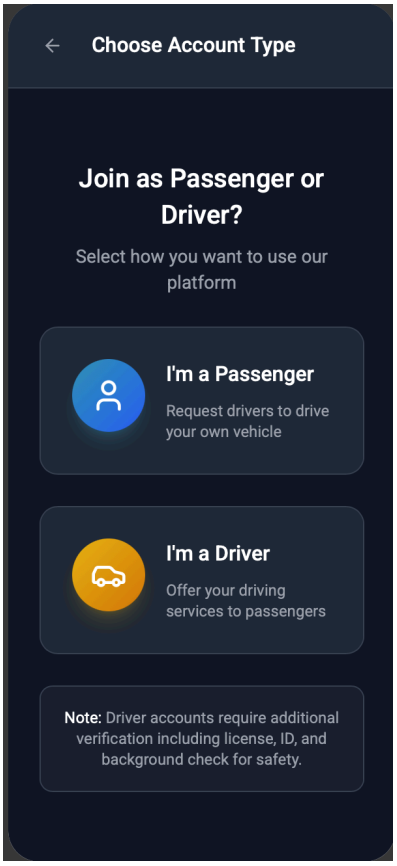


Fig. 15. Choosing Account type screen before proceeding to registration page.

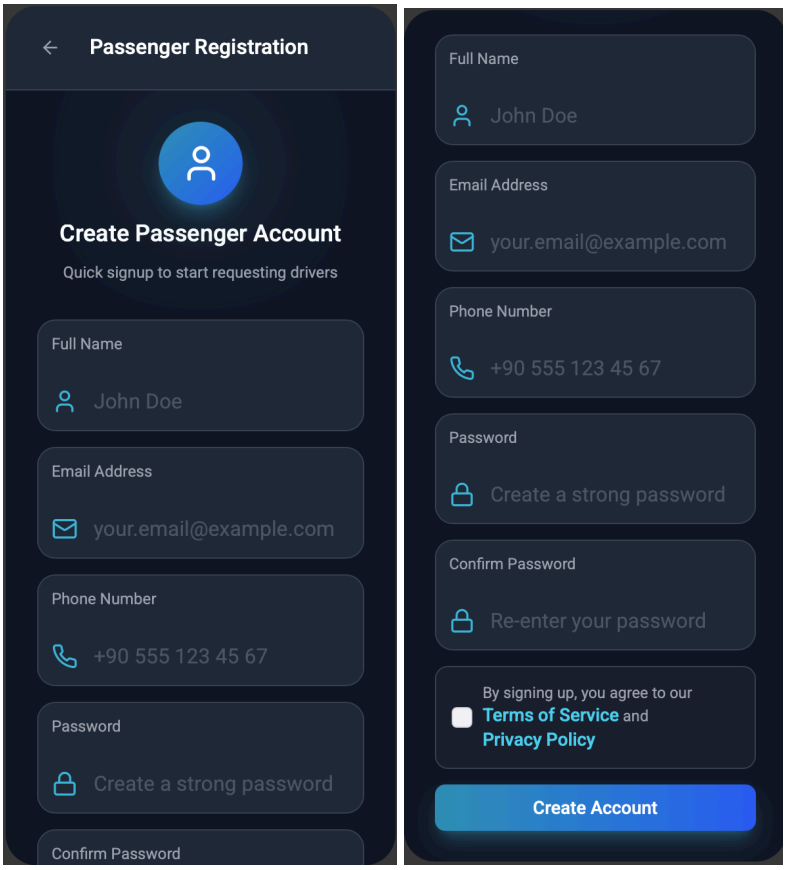


Fig. 16. Registration screen for passenger.

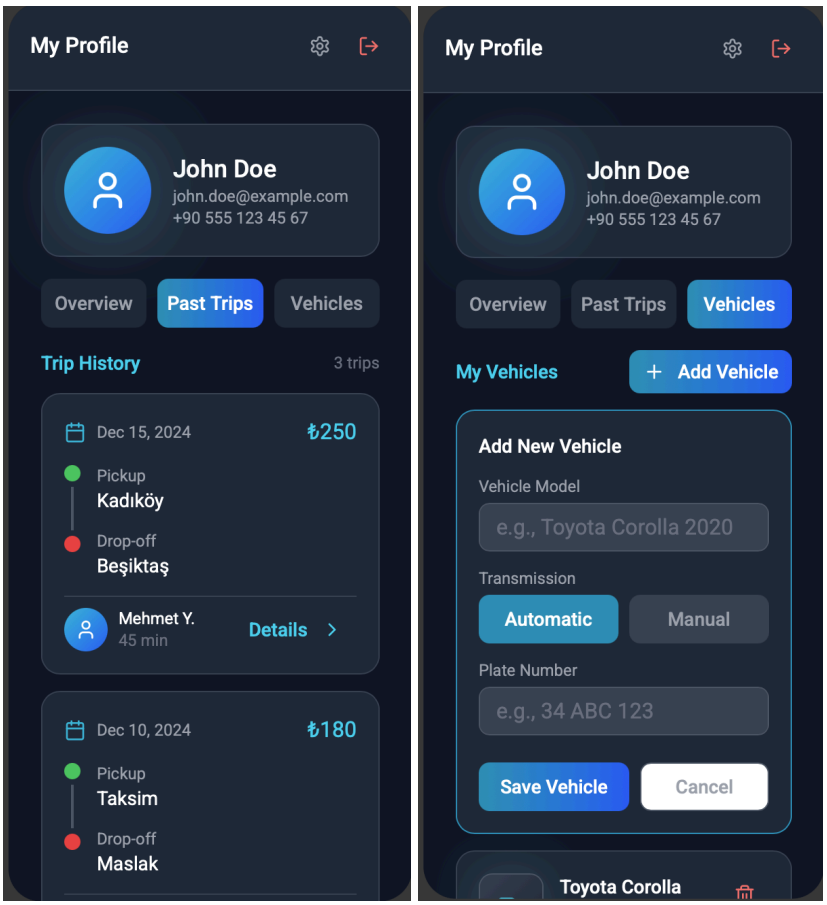


Fig. 17. Profile page for passenger.

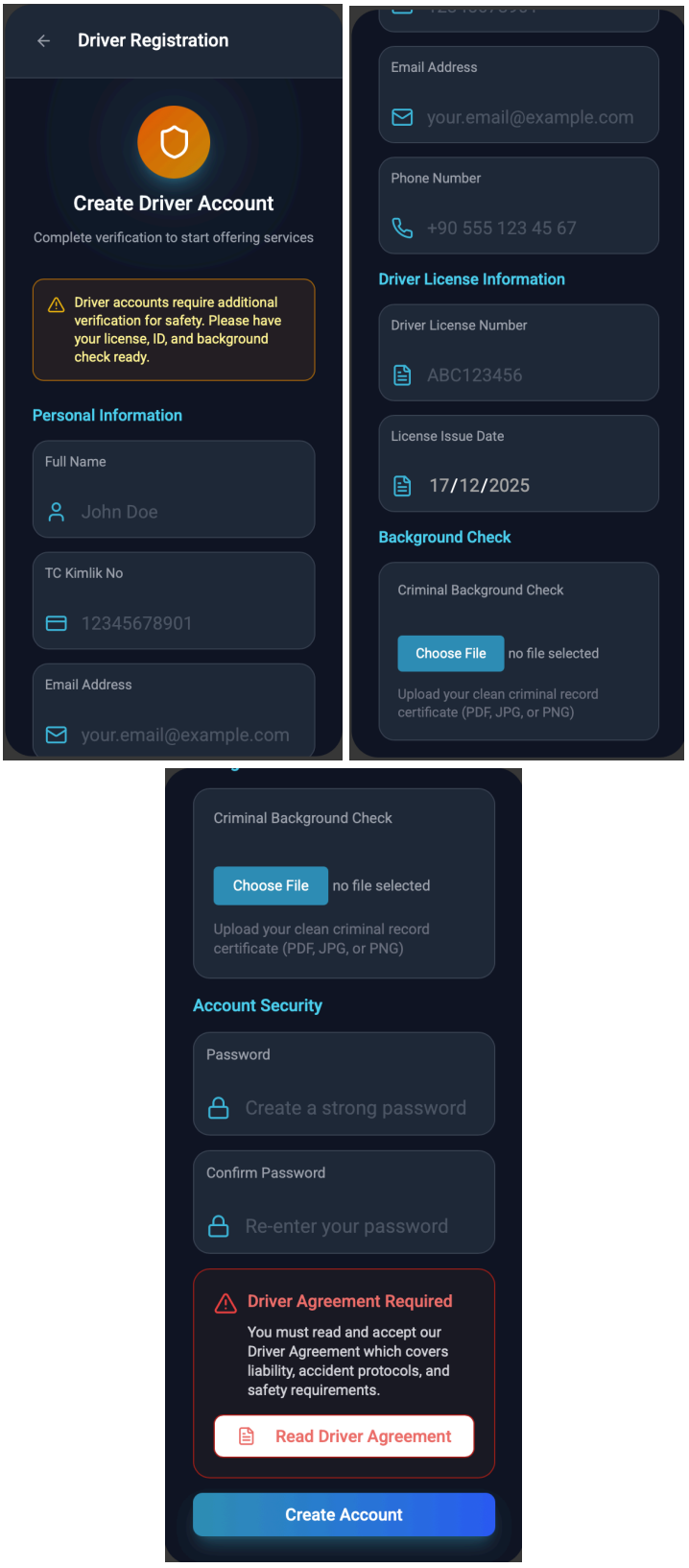


Fig. 18. Registration screen for driver.

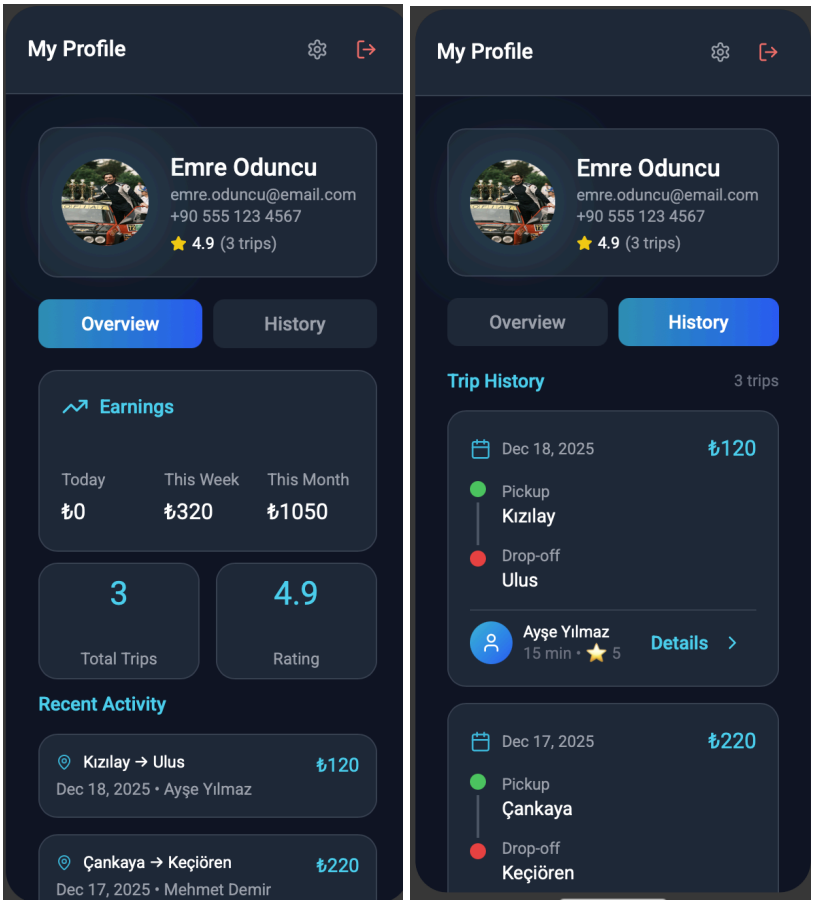


Fig. 19. Profile page for driver.

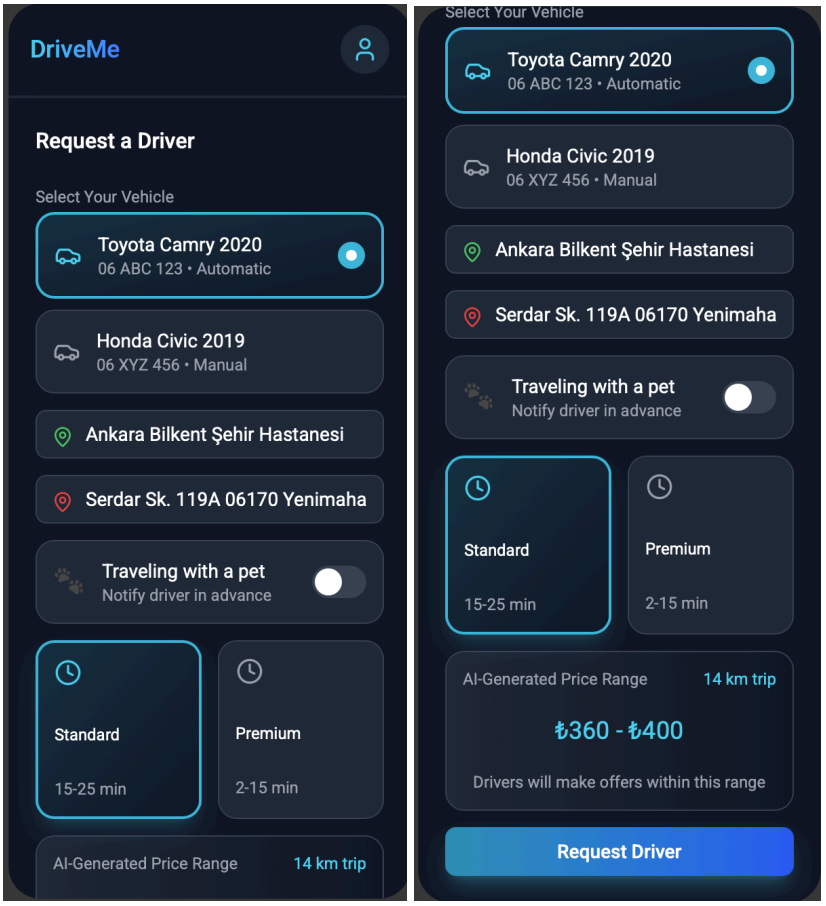
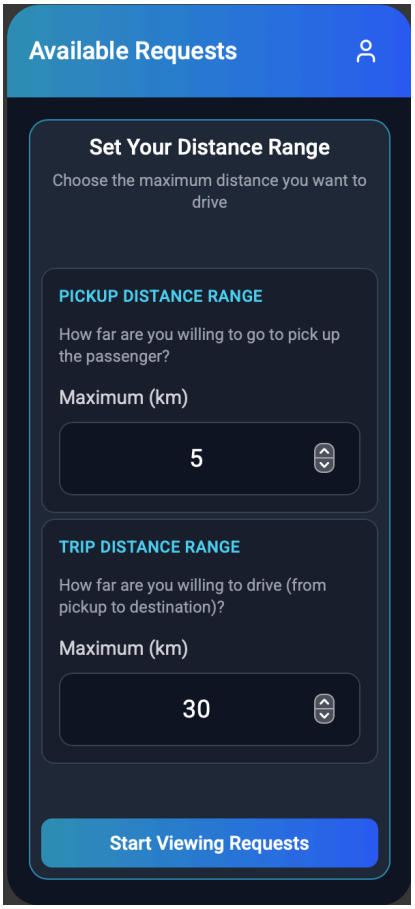
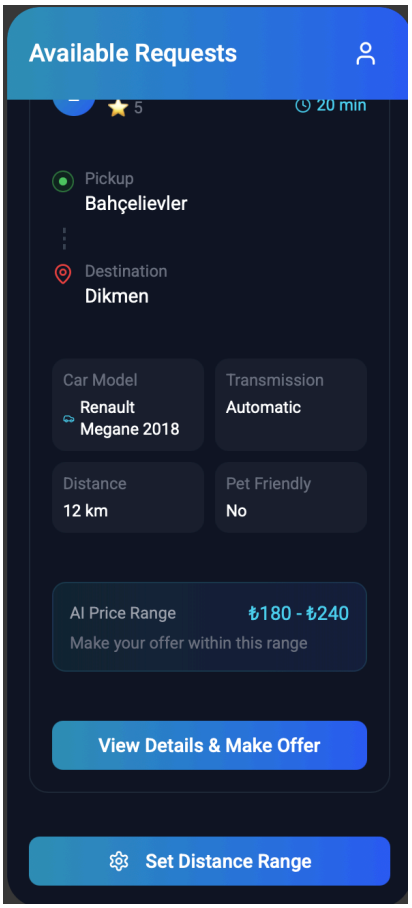
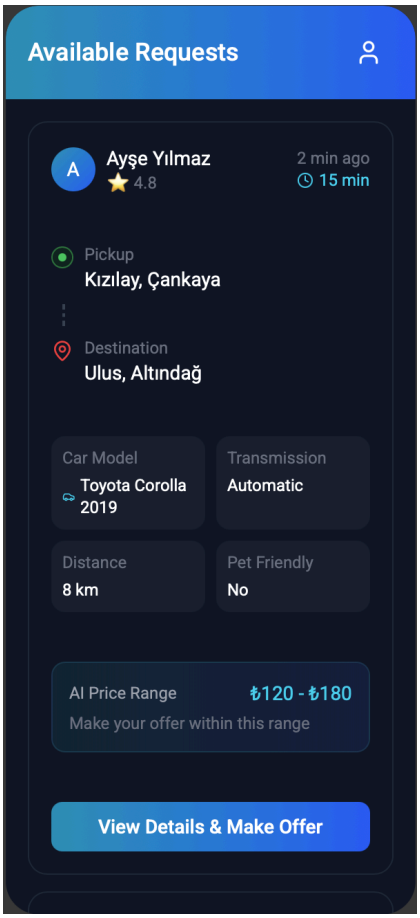


Fig. 20. Homepage for passenger.


Available Requests
Set Your Distance Range
Choose the maximum distance you want to drive
PICKUP DISTANCE RANGE
How far are you willing to go to pick up the passenger?
Maximum (km)
5
TRIP DISTANCE RANGE
How far are you willing to drive (from pickup to destination)?
Maximum (km)
30
Start Viewing Requests


Available Requests
5
20 min
Pickup
Bahçelievler
Destination
Dikmen
Car Model
Renault Megane 2018
Transmission
Automatic
Distance
12 km
Pet Friendly
No
AI Price Range
₺180 - ₺240
Make your offer within this range
View Details & Make Offer
Set Distance Range


Available Requests
A
Ayşe Yılmaz
4.8
2 min ago
15 min
Pickup
Kızılay, Çankaya
Destination
Ulus, Altındağ
Car Model
Toyota Corolla 2019
Transmission
Automatic
Distance
8 km
Pet Friendly
No
AI Price Range
₺120 - ₺180
Make your offer within this range
View Details & Make Offer

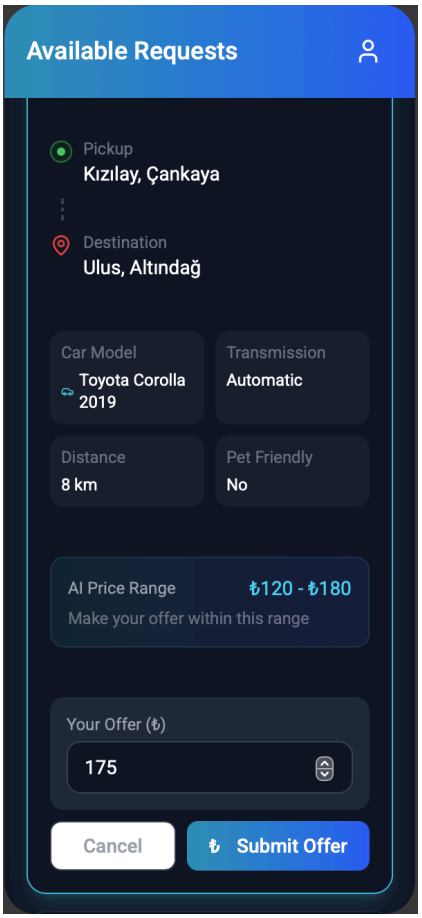

Available Requests
Pickup
Kızılay, Çankaya
Destination
Ulus, Altındağ
Car Model
Toyota Corolla 2019
Transmission
Automatic
Distance
8 km
Pet Friendly
No
AI Price Range
₺120 - ₺180
Make your offer within this range
Your Offer (₺)
175
Cancel
Submit Offer

Fig. 21. Homepage for driver.

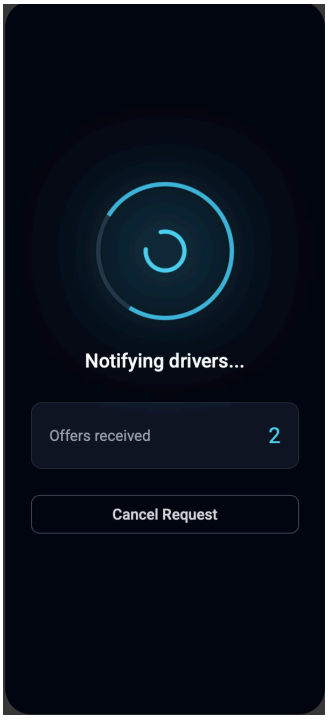


Fig. 22. Waiting offers screen for passenger.

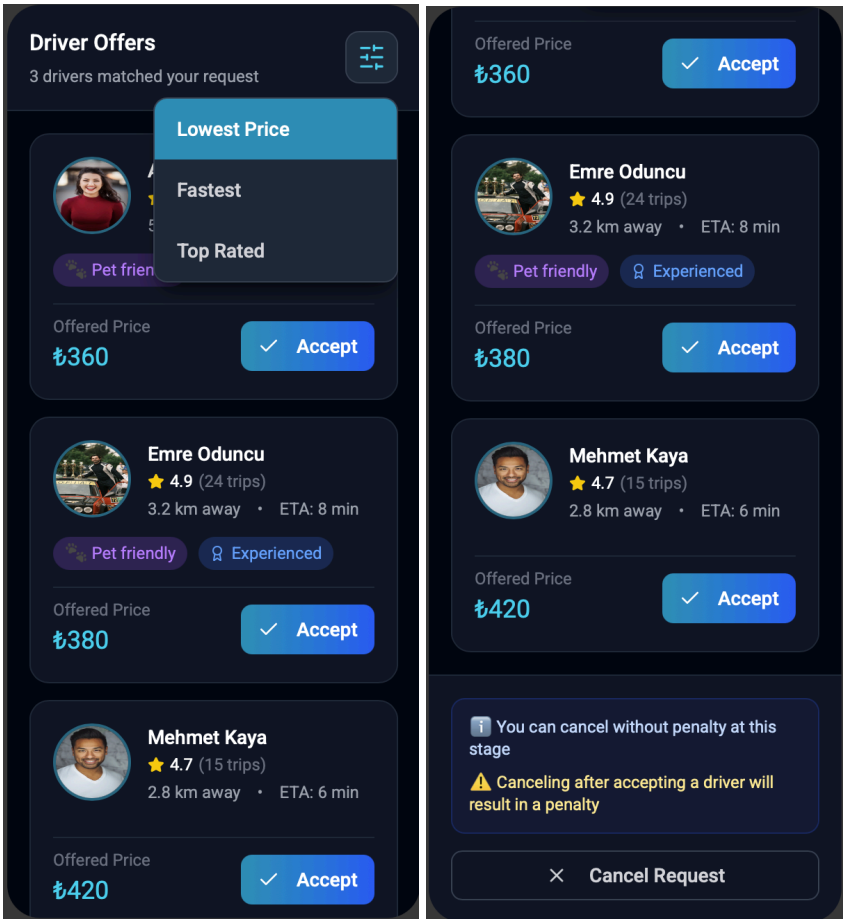


Fig. 23. Offers screen for passenger.

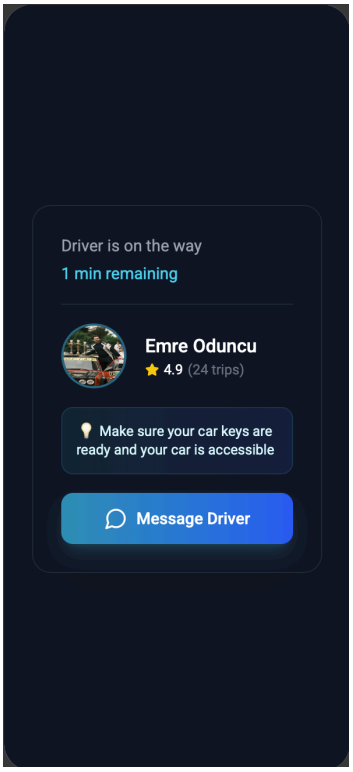


Fig. 24. Waiting for driver screen for passenger.

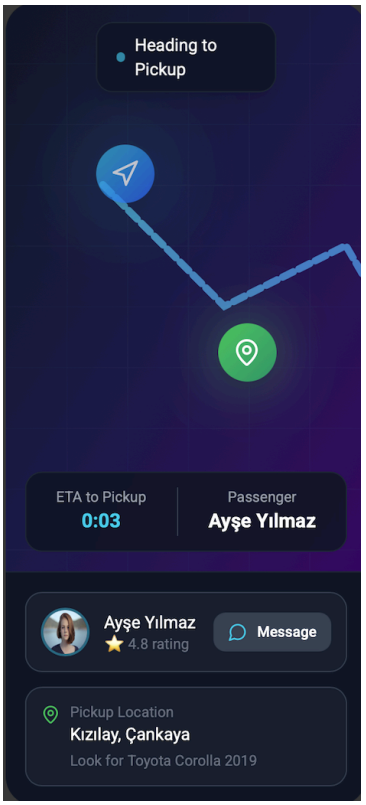


Fig. 25. Heading to Pickup screen for driver.

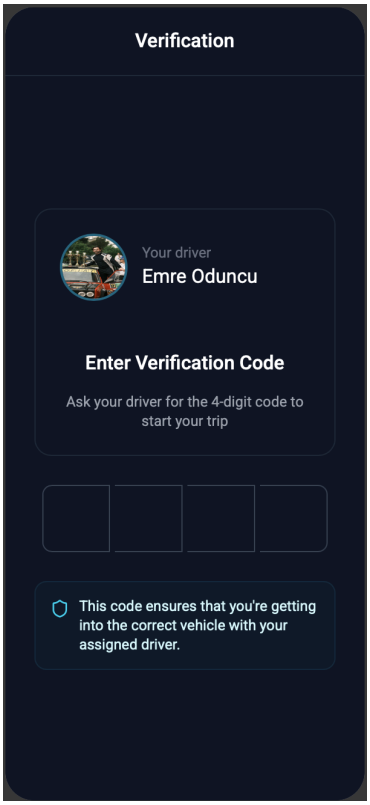


Fig. 26. Verification screen for passenger.

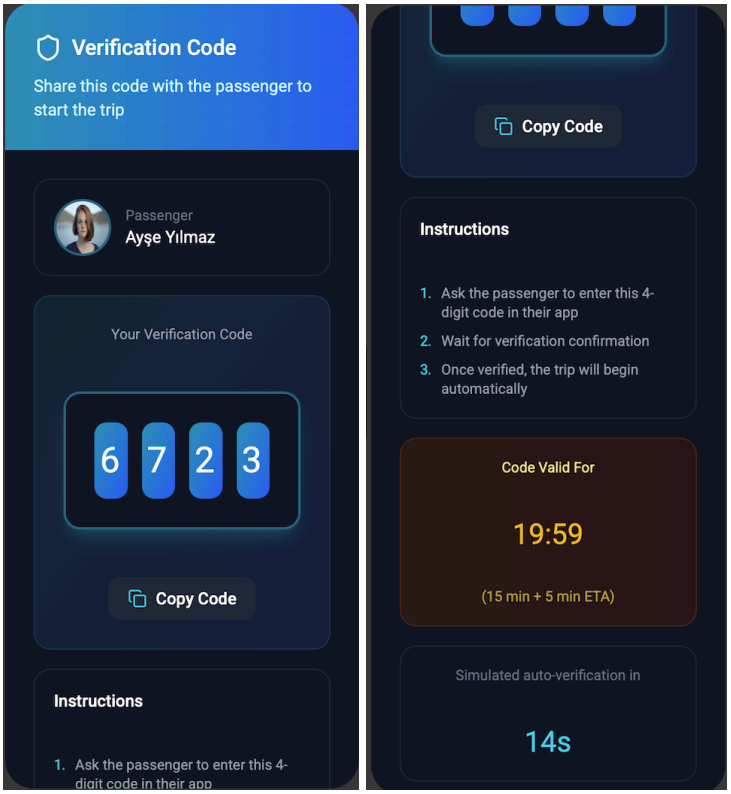


Fig. 27. Verification screen for driver.

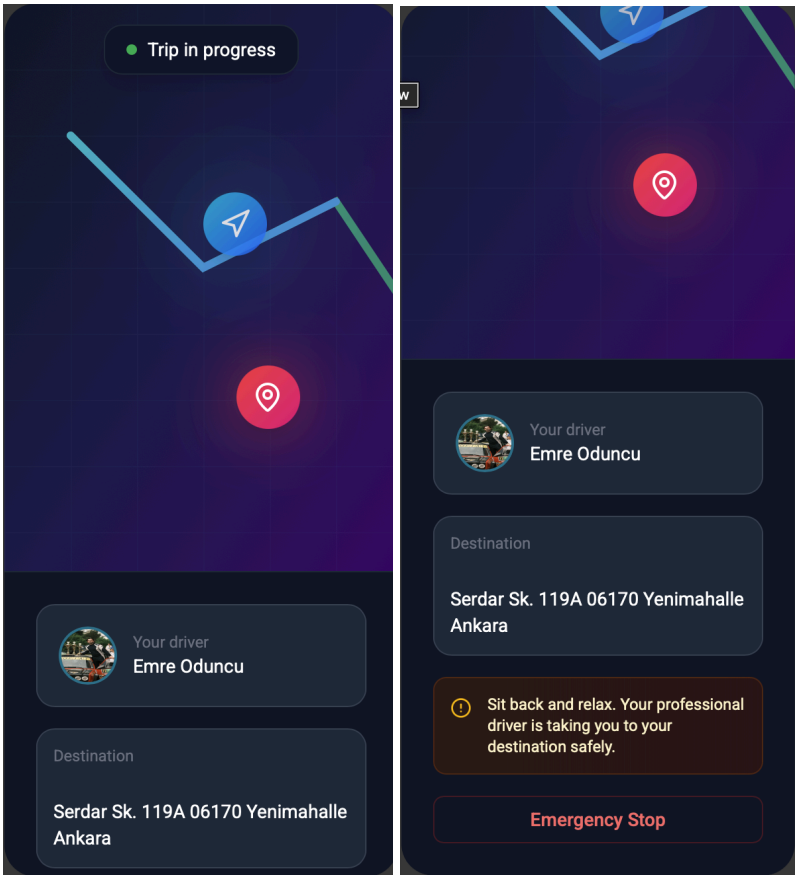


Fig. 28. Screen during trip for passenger.

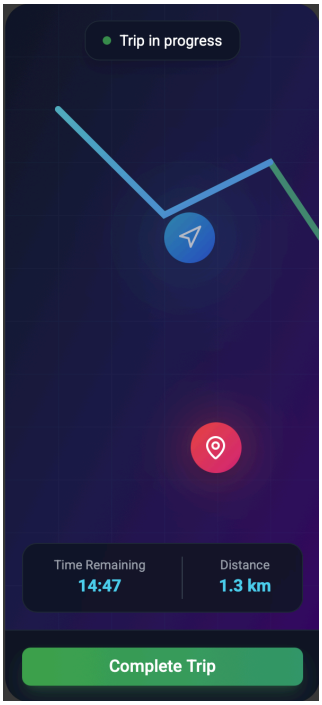


Fig. 29. Screen during trip for driver.

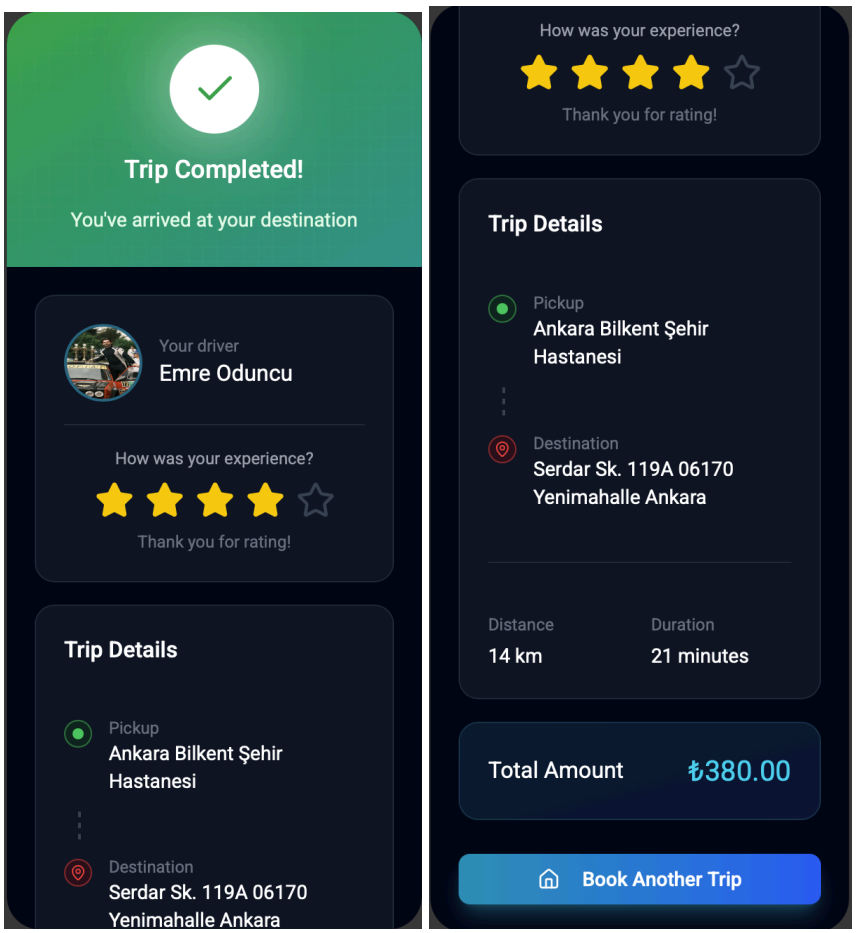


Fig. 30. Screen after trip is completed for passenger.

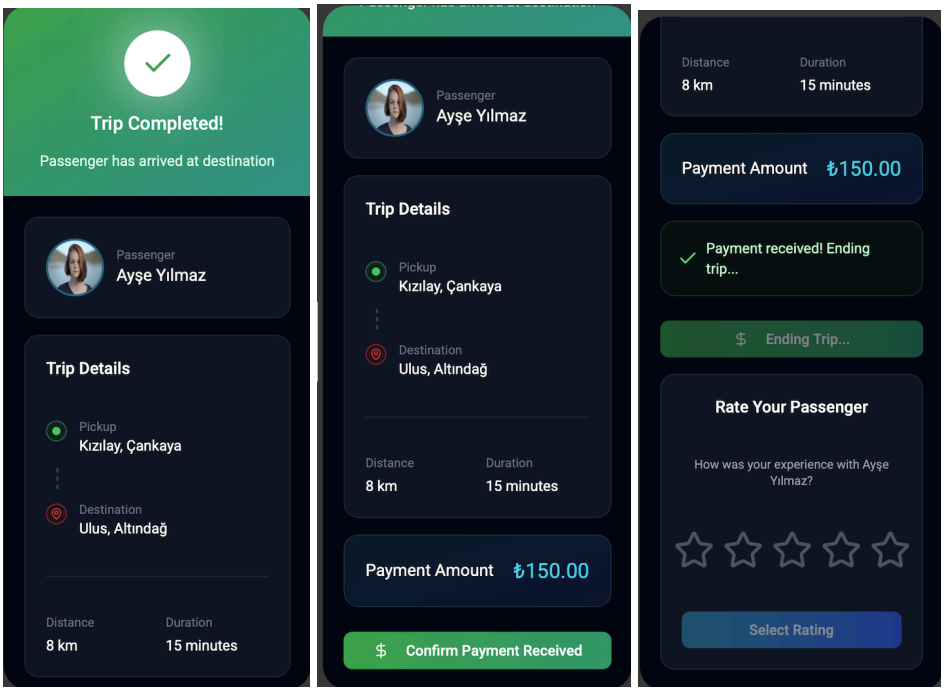


Fig. 31. Screen after trip is completed for driver/

4 Other Analysis Elements

4.1 Consideration of Various Factors in Engineering Design

4.1.1 Constraints

This section discusses the constraints of this project in detail. Since the practical aspects extend to the real world, solutions that are offered will most likely be affected by various factors. Below is a table that displays possible factors that might affect the design process.

TABLE II

FACTORS TABLE

	Effect Level	**Effect**
Global Factors	8	Take different road maps and documentation needs for different countries and regions into consideration.
Economical Factors	9	The price that the application offers shall be within a reasonable economical range and the local currency.
Public Safety	10	● The drivers that the app lists must be legally accountable for traffic safety. ● The vehicles must undergo proper maintenance and should be able to provide proof if necessary.
Environmental Factors	5	The roads that are taken or proposed by the application shall minimize gas consumption in an eco-friendly approach.

Accessibility	7	The application should be accessible to the elderly and the disabled since they might be possible users.

3.1.1.1 Developmental Constraints

- **No Online Payment Services:** Following CS491 guidelines, the system cannot integrate credit cards, digital wallets, or payment gateways; only cash transaction logging is permitted.
- **Mobile-Only Deployment:** The project scope limits functional deployment to mobile platforms (Android), restricting development of web-based front-end features.
- **Dependence on Third-Party APIs:** Geolocation and real-time tracking rely on external services (e.g., Google Maps). API limitations, outages, or pricing changes directly constrain system functionality.

3.1.1.2 Legal and Regulatory Constraints

- **Compliance with Turkish Personal Data Protection Law (KVKK):** Since the system processes personal data such as names, phone numbers, location information, and driver license photos, all database design, storage, and access-control mechanisms must comply with KVKK requirements. This imposes constraints on data retention, encryption, access permissions, and auditability.
- **Restrictions on Shared-Ride / Commercial Transport Services in Türkiye:** Turkish transportation regulations prohibit informal ride-sharing or peer-to-peer commercial passenger transport. Therefore, the DriveMe platform cannot support shared rides, pooled trips, or commercial taxi-like operations. The application must restrict itself to the allowed scenario of driver services where a driver operates the user's own car, not shared passenger transportation.

3.1.1.3 Technological Constraints

- **Internet Connection:** The application will need internet connection to function properly and in-real-time.
- **Location/GPS Access:** The application will need to access the location of the user in certain scenarios like the location tracking through the trip.
- **Device and OS Limitations:** The system must perform reliably on standard mid-range Android devices due to limited testing hardware availability. The user devices will need at least 4 GBs of RAM and Android 8.0+.

3.1.1.4 Sustainability Constraints

- **Database Enlargement:** In the case of unexpectedly high volume of users or vehicles the databases may need to be enlarged, which will cause a higher cost for renting servers.
- **Annual Fees:** The SAAS's and possible memberships, as well as app distribution service registrations like Google Play Store are needed to be paid each year.

4.1.2 Standards

The development of DriveMe should be conducted in a way that all the source code and documentation shall follow these standards:

- IEEE 1471 is a documentation standard that focuses on system architecture. It is a recommended framework for software systems like this as per its software-intensive structure. This style offers the ability to perceive and understand the architecture, its components, and the relationships between the components. It highlights the actors and different parties that need to be considered while making some critical decisions about the architecture and/or the system. The architectural documentation shall follow IEEE 1471 standard.
- UML is a standardized modeling language that is often used to create Object-Oriented documentation models for software systems. The

three types that are offered in UML are structural diagrams, behavioral diagrams and interactional diagrams. Each of these categories have various sub-types that may be useful for capturing different aspects and functions of a software system.

- ACM Code of Ethics and Professional Conduct should be followed by all engineers/developers to maintain a professional lifecycle.
- IEEE Citation Style should be used for each document and their references.

4.2 Risks and Alternatives

One of DriveMe's key goals is to deliver a continuous on demand driving experience by connecting vehicle owners with independent drivers through a real-time Matching Engine. To achieve this, various location-based applications and secure backends are used to ensure efficient driving coordination. However, this requires high-speed database responses and convergent network connectivity. One of the main risks is that the Matching Engine may not be able to deliver peak demand switching distribution lines accurately or quickly enough to be useful to drivers. In such a case, the system cannot provide bid options in a timely manner, leading to dissatisfaction for both parties. If it does not produce reliable results over the lifetime of the matching model, we plan to modify the architecture so that a molded localized caching is stored to ensure that expected requests appear even at small backend latencies.

TABLE III

RISK AND ALTERNATIVES

	Likelihood	**Effect on the project**	**B Plan Summary**
Matching Engine fails to broadcast requests	Medium	Drivers cannot see or bid on available trips.	Implementing a localized caching system for pending requests to ensure visibility even during minor backend lags.

Payment Gateway timeout during capture	Low	Financial transactions cannot be finalized.	Queuing failed transactions for allowing the user to provide an alternative payment method.
Pricing Engine calculation delay	Medium	Dynamic fares cannot be generated instantly.	Reverting to a base distance-only calculation based on the initial bid if real-time traffic data is unreachable.

4.3 Project Plan

TABLE IV

FACTORS

	Effect level	**Effect**
Public health	Medium	The system contributes indirectly to public health by reducing impaired driving risks, which influences safety-oriented requirements and ethical considerations in system design.
Public safety	High	Preventing drunk or fatigued driving is a core objective, directly shaping functional requirements, validation mechanisms, and reliability constraints.
Public welfare	Medium	Improved mobility access and safer transportation alternatives support societal welfare, affecting usability and accessibility requirements.
Global factors	Low	The project is primarily local and academic; however, scalability and future extensibility considerations influence architectural decisions.
Cultural factors	Medium	User trust, privacy sensitivity, and attitudes toward ride and driver services influence UX design and verification mechanisms
Social factors	High	User behavior, trust between drivers and car owners, and rating systems

		strongly affect interaction design and platform policies.
Environmental factors	Low	Environmental impact is indirect; however, reducing unnecessary vehicle abandonment may slightly contribute to reduced traffic congestion.
Economic factors	Medium	Development budget, cloud service costs, and API usage limits influence architectural trade-offs and feature prioritization.

TABLE V

LIST OF WORK PACKAGES

WP#	Work Package Title	Leader	Members Involved
WP1	Project Specification Document	Ufuk Baran Güler	All Members
WP2	Analysis and Requirement Report	Eda Alparslan	All Members
WP3	Frontend Development	Duru Solakoğlu	Berfin Çetinkaya, Ege Kaan Eren
WP4	Backend Development	Ege Kaan Eren	Ufuk Baran Güler, Duru Solakoğlu
WP5	Database Setup	Berfin Çetinkaya	Ege Kaan Eren
WP6	Recommendation System Development	Ufuk Baran Güler	Eda Alparslan, Berfin Çetinkaya
WP7	Demo	Duru Solakoğlu	All Members
WP8	Detailed Design Report	Eda Alparslan	All Members
WP9	Final Design Project Report	Berfin Çetinkaya	All Members
WP10	App Launch	Ege Kaan Eren	All Members
WP11	Final Demo	Ufuk Baran Güler	All Members

TABLE IV

DETAILED WPs

WP 1: *Project Specification Document*			
Start date: *November 2025* **End date:** *November 2025*			
Leader:	*Ufuk Baran Güler*	**Members involved:**	*All team members*
Objectives: The objective of this work package is to define the overall project vision, scope, and high-level requirements. It establishes the foundation for all subsequent analysis, design, and development activities. Ethical, professional, and feasibility considerations are identified and documented.			
Tasks: Task 1.1: Project Introduction: Describe the project background, motivation, objectives, and innovation aspects. Task 1.2: Requirements Definition: Identify high-level functional and non-functional requirements. Task 1.3: Constraints and Risks: Document assumptions, constraints, and initial project risks. Task 1.4: References: Collect and format all academic and technical references.			
Deliverables **D1.1:** Project Specification Document			
WP 2: *Analysis and Requirement Report*			
Start date: December 2025 **End date:** December 2025			
Leader:	*Eda Alparslan*	**Members involved:**	*All team members*
Objectives: This work package aims to analyze system requirements in detail and model the system using standard software engineering techniques. The goal is to ensure correctness, completeness, and consistency of requirements before implementation begins.			
Tasks: Task 2.1: Scenario definitions Task 2.2: Use-case diagram creation Task 2.3: Object and class model development Task 2.4: Dynamic modeling (activity, sequence, and state diagrams) Task 2.5: User interface mockups **Task 2.6:** Analysis of influencing factors, risks, and ethical responsibilities **Task 2.7:** References			
Deliverables D2.1: Analysis and Requirement Report			

WP 3: *Frontend Development*			
Start date: *December 2025* **End date:** *May 2026*			
Leader:	*Duru Solakoğlu*	**Members involved:**	*Berfin Çetinkaya, Ege Kaan Eren*

Objectives: The objective of this work package is to implement the user-facing components of the system according to the approved UI designs, ensuring usability, performance, and consistency.			
Tasks: Task 3.1: Login and registration pages Task 3.2: Onboarding interface Task 3.3: Main application screens Task 3.4: Navigation and search components Task 3.5: Frontend–backend integration Task 3.6: Performance and usability optimization			
Deliverables **D3.1:** Frontend application			
WP 4: *Backend Development*			
Start date: *December 2025* **End date:** *May 2026*			
Leader:	*Ege Kaan Eren*	**Members involved:**	*Ufuk Baran Güler, Duru Solakoğlu*
Objectives: This work package focuses on implementing backend services based on the system design. It ensures scalability, reliability, and secure communication with frontend components.			
Tasks: Task 4.1: Backend project setup Task 4.2: Core service and controller implementation Task 4.3: Integration with external services Task 4.4: API testing and validation Task 4.5: Deployment preparation			
Deliverables **D4.1:** Backend application			

WP 5: *Database Setup*			
Start date: *December 2024* **End date:** *February 2025*			
Leader:	*Berfin Çetinkaya*	**Members involved:**	*Ege Kaan Eren*
Objectives: The objective of this work package is to design and implement a database that supports system functionality, ensures data integrity, and enables efficient data retrieval.			
Tasks: Task 5.1: Database schema design Task 5.2: Table creation and normalization Task 5.3: Initial data population Task 5.4: Database performance validation			
Deliverables **D5.1:** Operational database schema			
WP 6: *Recommendation System Development*			
Start date: *January 2025* **End date:** *May 2025*			

Leader:	*Ufuk Baran Güler*	**Members involved:**	*Eda Alparslan, Berfin Çetinkaya*
Objectives: *This work package aims to design and implement a recommendation mechanism that provides personalized suggestions based on user behavior and preferences.*			
Tasks: Task 6.1: Define recommendation criteria Task 6.2: Weight assignment and scoring logic Task 6.3: Algorithm implementation and testing			
Deliverables **D6.1:** Recommendation system module			

WP 7: *Demo*			
Start date: *December 2025* **End date:** *February 2026*			
Leader:	*Duru Solakoğlu*	**Members involved:**	*All team members*
Objectives: The objective of this work package is to prepare and present a functional demonstration of the system for evaluation.			
Tasks: Task 7.1: Slide preparation Task 7.2: Demo scenario preparation Task 7.3: Presentation			
Deliverables **D7.1:** Demo presentation			
WP 8: *Detailed Design Report*			
Start date: *February 2026* **End date:** *March 2026*			
Leader:	*Eda Alparslan*	**Members involved:**	*All team members*
Objectives: This work package documents the detailed system design, including architecture, subsystems, and testing strategies.			
Tasks: Task 8.1: Architectural design Task 8.2: Subsystem definitions Task 8.3: Test case definitions			
Deliverables **D8.1:** Detailed Design Report			

WP 9: *Final Design Project Report*			
Start date: *April 2026* **End date:** *May 2026*			
Leader:	*Berfin Çetinkaya*	**Members involved:**	*All team members*
Objectives: The objective of this work package is to compile all project outcomes into a final academic report.			
Tasks: Task 9.1: Final documentation Task 9.2: Maintenance and limitation discussion Task 9.3: Final review			
Deliverables **D9.1:** Final Design Project Report			
WP 10: *App Launch*			
Start date: *May2025* **End date:** *May 2025*			
Leader:	*Ege Kaan Eren*	**Members involved:**	*All team members*
Objectives: This work package ensures that the application is fully tested and ready for deployment.			
Tasks: Task 10.1: Final system testing Task 10.2: Deployment preparation Task 10.3: Release execution			
Deliverables **D10.1:** Deployed application			

WP 11: *Final Demo*			
Start date: *May2025* **End date:** *May 2025*			
Leader:	*Ufuk Baran Güler*	**Members involved:**	*All team members*
Objectives: The objective of this work package is to deliver the final demonstration and formally conclude the project.			
Tasks: Task 11.1: Final slide preparation Task 11.2: Final demo execution Task 11.3: Project presentation			
Deliverables **D11.1:** Final Demo Presentation			

4.4 Ensuring Proper Teamwork

- Task distributions are decided jointly by group. Daily updates on progress and possible obstacles are shared between members.
- In cases where consensus cannot be reach ed regarding technical decisions or project management, the matter will be discussed professionally until a mutual conclusion is reached.

4.5 Ethics and Professional Responsibilities

- The DriveMe platform involves a number of ethical and professional responsibilities given the way it processes personal data and third-party drivers who handle users' private vehicles. During the analysis and design phases, special consideration was given to user safety, data privacy, transparency, and fairness.
- DriveMe processes a wide range of sensitive information: identity data, vehicle data, and geolocation. According to KVKK in Turkey and GDPR (General Data Protection Regulation) in the European Union, only data that is strictly necessary for core system functionality is collected. Location data is utilized strictly for driver-passenger matching, navigation, and trip verification. Location data is not stored for longer than operationally necessary. Users are informed about the usage of data through agreements presented at account creation.
- Professional responsibility is further emphasized by well-defined roles and accountability. Drivers accept digital agreements that define their responsibilities pertaining to safe driving behavior, traffic violations, and cancellations. Besides that, dynamic pricing ranges and penalty mechanisms are introduced to promote fairness in pricing and avoid unsafe or irresponsible behavior.
- From the perspective of software engineering, the project has pursued ethical development practices: source code will remain private during the whole duration of the project, and it has taken into consideration the licensing implications of all third-party libraries and frameworks.

4.6 Planning for New Knowledge and Learning Strategies

- The creation of the DriveMe platform will entail learning new technology skills that surpass the current expertise that the team has. Due to the involvement of the project in managing trips in real-time, geolocation, secure authentication, and the use of AI technology pricing engines, there are areas that the team has identified in the learning phase.
- The initial major learning domain is that of integration and real-time systems with mobile and backend. The team aims to enhance their understanding about RESTful API design, communication over WebSocket, and location-based services by reading up on documentation available in the framework and developing prototypes on a small scale. The learning approach in this domain is by developing.
- The next area where learning is required is that of system security and authentication. In order to facilitate safe user login and authentication, the team also intends to learn about OAuth2 and JWT tokens, focusing on how these tokens work as a form of authentication and implement it directly into their project.
- It also comprises AI-inclusive pricing and decision-making. Though the team has experience in basic concepts related to algorithms, additional learning needs to happen to develop a fair and transparent pricing mechanism. Such expertise would be obtained by reading online publications, academic papers, and practicing basic concepts related to pricing.
- Moreover, their aim would also be to enhance their knowledge regarding deployment as well as cloud infrastructure. Cloud hosting services and the resilience of the system would be considered by referring to the tutorials offered by cloud services.
- Peer learning and sharing of tasks will be employed to share the knowledge among the group.

5 Glossary

- **AI:** Artificial intelligence

- **RESTful API:** It is an application programming interface that follows the REST architectural principles by providing stateless, resource-oriented communication between clients and servers using standard HTTP methods and status codes.
- **UML**: Unified Modelling Language.
- **IEEE**: Institute of Electrical and Electronics Engineers. A professional association that also creates frameworks and documentation guidelines.
- **GPS:** Global Positioning System. A satellite-based navigation system.
- **OS:** Operating system. In this document it usually refers to Android.
- **KVKK:** Turkish Personal Data Privacy Law.
- **SAAS:** Software as a service.
- **GDPR:** General Data Protection Regulation by the European Union.
- **OAuth2.0:** An industry-standard protocol for authorization.
- **JWT:** JSON Web Token

6 References

[1] Dryver Inc., “Dryver – Personal Driver Service,” *Dryver Official Website*. [Online]. Available: https://www.dryver.com. Accessed: Sep. 2025.

[2] Get Driven LLC, “Get Driven – On-Demand Personal Driver App,” *Get Driven Official Website*. [Online]. Available: https://www.getdriven.app. Accessed: Sep. 2025.

[3] Cab Istanbul, “Chauffeur and Private Driver Services in Türkiye,” *Cab Istanbul Official Website*. [Online]. Available: https://cabistanbul.com/service/driver-services. Accessed: Sep. 2025.

[4] Wikipedia contributors, “DiDi,” *Wikipedia, The Free Encyclopedia*. [Online]. Available: https://en.wikipedia.org/wiki/DiDi. Accessed: Sep. 2025.

[5] Figma Inc., “Figma – Collaborative Interface Design Tool,” *Figma*. [Online]. Available: https://www.figma.com. Accessed: Sep. 2025.

[6] Visual Paradigm Online, “Visual Paradigm Online – UML and System Modeling Tool,” *Visual Paradigm*. [Online]. Available: https://online.visual-paradigm.com. Accessed: Sep. 2025.

[7] B. Bruegge and A. H. Dutoit, *Object-Oriented Software Engineering: Using UML, Patterns, and Java*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2004, ISBN: 0-13-047110-0.